

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-125773

(43)Date of publication of application : 11.05.2001

---

(51)Int.Cl. G06F 9/06

G06F 13/00

G06F 13/10

G06F 13/14

---

(21)Application number : 11-310010 (71)Applicant : SHARP CORP

(22)Date of filing : 29.10.1999 (72)Inventor : TANAKA YOSHIRO

---

(54) PERIPHERAL DEVICE FOR INFORMATION PROCESSORS CONNECTED BY NETWORK AND VERSION MANAGING METHOD FOR DEVICE DRIVER

(57)Abstract:

PROBLEM TO BE SOLVED: To efficiently and reliably perform device driver updating for a plurality of information processors connected by a network to use peripheral devices.

SOLUTION: A plurality of host computers 1 are connected through an intranet 6 and can share a digital compound machine 2 as a peripheral device. Each of host computers 1 is provided with a printer driver 12 and a scanner driver 13 as the device driver of an OS 11 for utilizing the digital compound machine 2. The version information of these device drivers is managed by a version information transmitting/receiving part 25 of the digital compound machine 2 and on the basis of the control by a time managing part 26, the inquiry of version information is performed

to a device driver managing server computer 3 at fixed timing. The inquiry result of the version information is reported through the intranet 6 to each of host computers 1.

---

LEGAL STATUS [Date of request for examination] 01.02.2002

[Date of sending the examiner's decision of rejection] 12.07.2005

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3764311

[Date of registration] 27.01.2006

[Number of appeal against examiner's decision of rejection] 2005-15485

[Date of requesting appeal against examiner's decision of rejection] 11.08.2005

[Date of extinction of right]

**\* NOTICES \***

JPO and NCIP are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.\*\*\*\* shows the word which can not be translated.

3.In the drawings, any words are not translated.

---

**CLAIMS**

---

[Claim(s)]

[Claim 1] In the peripheral device controlled by the device driver which two or more information processors are connected through an internal network, and is installed in those information processors It connects with the management server equipment which manages this device driver through the network to the exterior. An inquiry means to perform the inquiry of the version information of the device driver currently held at this management server equipment, and to acquire version information, The peripheral device of the information processor which is characterized by including a notice means to notify the version information acquired by the inquiry means to said information processor and by which network connection was carried out.

[Claim 2] Said inquiry means is the peripheral device of the information processor according to claim 1 which is characterized by having the timer means to which querying of the version information of a device driver is carried out by fixed time amount to said server equipment and by which network connection was carried out.

[Claim 3] Said inquiry means is the peripheral device of the information processor according to claim 1 which is characterized by having a spacing acquisition means to gain spacing which should carry out the inquiry of the version information of a device driver to next time, and a spacing modification means to make spacing at which this inquiry means performs an inquiry change into spacing gained by the spacing acquisition means and by which network connection was carried out to said management server equipment.

[Claim 4] Said inquiry means is the peripheral device of the information processor according to claim 1 which is characterized by detecting power-source starting and having a starting detection means to make the version information of a device driver asked only at the time of starting and by which network connection was carried out to said server equipment.

[Claim 5] A version storing means to store the version information of the device driver which said information processor is using, A version judging means to judge the difference with the version information stored in the version storing means, and the version information of the device driver obtained by the inquiry to the management server equipment by said inquiry means, Only when the judgment result by the version judging means has a difference, this inquiry \*\*\*\*\* version information The peripheral device of the information processor according to claim 1 to 4 which is characterized by having the notice control means which an information processor is made to notify with said notice means and by which network connection was carried out.

[Claim 6] The peripheral device of the information processor according to claim 1 to 5 which is characterized by having a version control means to make only the information processor which receives a notice demand of said version information and has a demand in said notice means from said information processor notify version information and by which network connection was carried out.

[Claim 7] A preservation means by which said device driver can be saved, and a version storing means to store the version information of the device driver which said information processor is using, A version judging means to judge the difference with the version information stored in said version storing means, and the version information of the device driver obtained by the inquiry to the management server equipment by said inquiry means, The peripheral device of the information processor according to claim 1 to 6 which is characterized by having a download means to download this device driver for a preservation means from this management server equipment when the judgment result by the version judging means has a difference

and by which network connection was carried out.

[Claim 8] Said notice means is the peripheral device of the information processor according to claim 7 which is characterized by to perform the inquiry of whether to perform install of the device driver downloaded with said download means to this information processor when the difference with the version information of the device driver which the information processor which is as a result of [ by said version judging means ] a judgment, and is stored in said version storing means is using, and the version information of the device driver obtained from management server equipment by said inquiry means arises and by which network connection was carried out.

[Claim 9] Said download means is the peripheral device of the information processor according to claim 7 or 8 which is characterized by to download each file which constitutes a device driver from the address corresponding to each file held by the address maintenance means and by which network connection was carried out, including further an address maintenance means hold the address of two or more management server equipments in which the file whose device driver of said is constituted by multiple files and constitutes a device driver is stored corresponding to a file.

[Claim 10] Said inquiry means is the peripheral device of the information processor according to claim 1 to 9 which is characterized by accessing this management server equipment and by which network connection was carried out using the log in password of the address of the management server equipment set up beforehand.

[Claim 11] Said inquiry means is the peripheral device of the information processor according to claim 10 which is characterized by having a password acquisition means to gain the log in password in the case of accessing next time from said management server equipment, and a password maintenance means to hold the log in password gained by the password acquisition means and by which network connection was carried out.

[Claim 12] An investigation program acquisition means by which said inquiry means acquires OS environmental research program from said management server equipment, An investigation program sending means to send OS environmental research program which answered the demand from said information processor and the investigation program acquisition means acquired to this information processor, A results-of-an-investigation acquisition means to gain the results of an investigation by this OS environmental research program from this information processor, A results-of-an-investigation sending means to send the results of an investigation gained by the results-of-an-investigation acquisition means to this management server equipment is included. Said download means The peripheral device of the information processor according to claim 7 or 8 which is characterized by downloading the device driver offered from this management server equipment corresponding to the results of an investigation sent by the results-of-an-investigation sending means

and by which network connection was carried out.

[Claim 13] Said download means is the peripheral device of the information processor according to claim 7 or 8 which is characterized by to download the device driver of the version chosen by the version selection means from management server equipment and by which network connection was carried out including a version information acquisition means acquire the version information of the device driver of two or more versions by which said inquiry means is saved to said management server equipment, and a version selection means choose the version of the device driver which makes it install in said information processor.

[Claim 14] Said preservation means is the peripheral device of the information processor according to claim 12 or 13 which is characterized by saving the device driver for two or more OS's and by which network connection was carried out.

[Claim 15] Said download means is the peripheral device of the information processor according to claim 7 to 14 which is characterized by having a decryption means to download the device driver to which cipher processing was performed from said management server equipment, and to decode this code and by which network connection was carried out.

[Claim 16] Said inquiry means is the peripheral device of the information processor according to claim 7 to 15 which is characterized by having a file list receiving means to receive the file list corresponding to the configuration information of the device concerned by which a configuration information transmitting means to transmit to said management server equipment, and the configuration information transmitting means transmitted the configuration information of the peripheral device concerned from this management server equipment and by which network connection was carried out.

[Claim 17] Said inquiry means is the peripheral device of the information processor according to claim 1 to 16 which is characterized by notifying OS information on a device driver and the version information of this device driver which were installed in said host equipment at the time of the inquiry to said management server equipment to this management server equipment and by which network connection was carried out.

[Claim 18] Said download means is the peripheral device of the information processor according to claim 7 to 17 which is characterized by answering the detection result of an access-less detection means, and starting download of said device driver and by which network connection was carried out including an access-less detection means to detect that there is no predetermined time access from said information processor.

[Claim 19] Said inquiry means is the peripheral device of the information processor according to claim 1 to 18 which is characterized by asking the version information of a device driver using HTTP and by which network connection was carried out to said server equipment.

[Claim 20] Said notice means is the peripheral device of the information processor

according to claim 1 to 19 which is characterized by performing the demand and notice to said information processor using HTTP and by which network connection was carried out.

[Claim 21] In the approach for managing the version of the device driver installed in each information processor in order to connect two or more information processors to a peripheral device through an internal network and to control this peripheral device This peripheral device is connected to the management server equipment which manages this device driver through the network to the exterior. The version control approach of the device driver characterized by notifying the version information which performed the inquiry of the version information of the device driver currently held at this management server equipment from this peripheral device, and was acquired from this management server equipment to each \*\*\*\*\* from this peripheral device.

---

## DETAILED DESCRIPTION

---

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to the management method of the device driver for using peripheral devices, such as a printer connected with information processors, such as a personal computer and a workstation, through a network, and the peripheral device concerned with an information processor.

[0002]

[Description of the Prior Art] Information processors, such as a personal computer and a workstation, connect various peripheral devices, and they consist of the former so that it may become possible to extend the applicability as an information processor. The I/O device which inputs various information or outputs the information acquired from an information processor to an information processor as a peripheral device is used typically. In the information processor, operation system (it is hereafter called "OS" for short) takes charge of fundamental I/O for human being to use an information processor, and it is made to perform various application software on OS in many cases. OS performs processing about different hardware for each information processor of every, and application software is because effect of a difference of the hardware of each information processor can be made hard to be influenced using the function which OS offers.

[0003] However, the peripheral device connected to an information processor is various, and in order to operate those peripheral devices efficiently, it is necessary to use the device driver which is the drive program currently created so that each

peripheral device may be suited. Since huge storage capacity is required, it is not practical that OS is equipped with all the device drivers of the peripheral device which may be connected to an information processor. Generally, in case a peripheral device is connected, the method of performing install incorporating a required device driver is adopted.

[0004] The peripheral device itself stores the device driver for using the peripheral device in JP,8-161250,A, and the configuration which installs the device driver stored in the host computer which is the information processor connected to the peripheral device is proposed. Moreover, in JP,10-301884,A, it has the program which investigates the OS environment of a host computer with the device driver in the peripheral device, and the configuration which transmits the device driver suitable for the environment of the hardware of a host computer or software to a host computer from a peripheral device is indicated.

[0005]

[The technical problem which is going to solve a technical problem] Also once the program which a computer uses is created, in order to discover fault, to perform the correction or to improve the engine performance, amelioration is performed in many cases. Since version control is generally made by software and modification is reflected in a version, using the software of the newest version as much as possible is recommended.

[0006] The device driver of a peripheral device is also offered by the manufacturer of a peripheral device etc. for OS currently generally used widely, and the version control accompanying amelioration is also made. If amelioration of a program etc. is performed about a device driver, it being sent to a user or providing for a user through networks, such as the Internet, in the condition of having been recorded on record media, such as a floppy disk and CD-ROM, is performed.

[0007] In order to use one peripheral device together with two or more information processors in recent years, the configuration connected through a network is used. It is necessary to install the device driver for using a peripheral device in OS of each information processor in such network configuration. If amelioration is performed to the device driver to the same peripheral device, the device driver of a different version from the device driver till then needs to be installed by each information processor. Conventionally, generally, although the manager of a network system reinstalls the device driver of the version improved by each information processor, the time and effort of the management will become great. In advanced technology, such as JP,8-161250,A and JP,10-301884,A, a device driver can be automatically supplied and installed in a host computer from a peripheral device.

[0008] However, since the new goods based on version up or a new concept are released, the situation where it becomes impossible to correspond only by the device driver currently beforehand prepared for the peripheral device will produce software,

such as a device driver and OS.

[0009] The purpose of this invention is offering the peripheral device of the information processor which can manage efficiently the device driver installed in the information processor and by which network connection's was carried out, and the management method of a device driver corresponding to the evolution of the environment of the hardware of an information processor, or software used as version up and the host computer of a device driver.

[0010]

[Means for Solving the Problem] In the peripheral device controlled by the device driver by which two or more information processors are connected through an internal network, and this invention is installed in those information processors It connects with the management server equipment which manages this device driver through the network to the exterior. An inquiry means to perform the inquiry of the version information of the device driver currently held at this management server equipment, and to acquire version information, It is the peripheral device of the information processor which is characterized by including a notice means to notify the version information acquired by the inquiry means to said information processor and by which network connection was carried out.

[0011] If this invention is followed, two or more information processors will be connected to a peripheral device through an internal network. Each information processor controls a peripheral device by the device driver installed. A peripheral device includes an inquiry means and a notice means. It connects with the management server equipment which manages a device driver through the network to the exterior, and an inquiry means asks the version information of the device driver currently held at management server equipment, and acquires version information. A notice means notifies the version information acquired by the inquiry means to an information processor. The user using each information processor can judge whether the device driver which the information processor concerned is using is the newest version.

[0012] Moreover, by this invention, said inquiry means is characterized by having the timer means to which querying of the version information of a device driver is carried out by fixed time amount to said server equipment.

[0013] If this invention is followed, since a timer means is made to perform the inquiry of the version information of the device driver to the management server equipment by the inquiry means with a fixed time interval, an inquiry can be performed at a time etc. at a fixed time interval, for example, daily scheduled time and a daily week, and the user of an information processor can supervise continuously whether the device driver installed is the newest thing.

[0014] Moreover, by this invention, said inquiry means is characterized by having a spacing acquisition means to gain spacing which should carry out the inquiry of the



version information of a device driver to next time, and a spacing modification means to make spacing at which this inquiry means performs an inquiry change into spacing gained by the spacing acquisition means to said management server equipment.

[0015] If this invention is followed, since a spacing modification means changes the inquiry of the version information to the management server equipment by the inquiry means according to spacing which a spacing acquisition means gains from management server equipment, when the schedule of version up of a device driver has become clear beforehand by the management server equipment side, spacing is made to adjust according to a schedule and the inquiry doubled with the schedule of version up can be performed appropriately. Since the inquiry to management server equipment disappears from a peripheral device through an external network to the timing to which version up is not performed, the traffic to management server equipment is mitigable.

[0016] Moreover, by this invention, to said server equipment, said inquiry means detects power-source starting, and is characterized by having a starting detection means to make the version information of a device driver asked only at the time of starting.

[0017] Since an inquiry means will perform the inquiry of version information to management server equipment when the starting detection means has detected power-source starting if this invention is followed, a peripheral device can ask version information during the warming up accompanying power-source starting to management server equipment. Since there is little possibility that use of the peripheral device from two or more information processors connected to a network will be performed during warming up, the inquiry of version information can be performed to the timing which does not affect the effectiveness as the whole network etc. Since detection of starting of a power source can be performed with a comparatively easy configuration, version information is manageable by low cost from management which performs an inquiry with a fixed time interval.

[0018] Moreover, a version storing means to store the version information of the device driver for which said information processor is using this invention, A version judging means to judge the difference with the version information stored in the version storing means, and the version information of the device driver obtained by the inquiry to the management server equipment by said inquiry means, Only when the judgment result by the version judging means has a difference, it is characterized by having the notice control means to which an information processor is made to notify this inquiry \*\*\*\*\* version information with said notice means.

[0019] A version judging means judge the difference with a version storing means store the version information of the device driver which two or more information processors connected to a peripheral device through a network will use if this invention is followed, the version information stored, and the version information of

the device driver obtained by the inquiry to management server equipment, and the notice control means, to which an information processor makes notify the version information asked and acquired only when a judgment result had a difference with a notice means contain. Only when the version information of the device driver which the information processor is using is stored beforehand, and is managed, the difference with the version information of the device driver saved to management server equipment is judged and there is a difference, the traffic of the notice through an internal network can be reduced by notifying to an information processor.

[0020] Moreover, this invention is characterized by having a version control means to make only the information processor which receives a notice demand of said version information and has a demand in said notice means from said information processor notify version information.

[0021] If this invention is followed, from the information processor connected through a network, a peripheral device receives a notice demand of version information, and has a version control means to make only an information processor with a demand notify version information, to a notice means. Since version information is notified only to an information processor with a notice demand, as compared with notifying version information, the traffic in an internal network environment can be reduced to many and unspecified information processors. The need for version up is the information processor which newly introduces application software, and especially the notice of version information also has many unnecessary things at the information processor which the notice of version information may be required by the application software, and does not introduce such application software. Since version information is not notified to an information processor without the need for version up, mitigation of the amount of traffic can be aimed at.

[0022] Moreover, a preservation means by which this invention can save said device driver, A version storing means to store the version information of the device driver which said information processor is using, A version judging means to judge the difference with the version information stored in said version storing means, and the version information of the device driver obtained by the inquiry to the management server equipment by said inquiry means, When the judgment result by the version judging means has a difference, it is characterized by having a download means to download this device driver for a preservation means from this management server equipment.

[0023] A preservation means by which a peripheral device can save a device driver if this invention is followed, A version storing means to store the version information of the device driver which the information processor is using, A version judging means to judge the difference with the version information stored and the version information acquired by the inquiry to management server equipment, Since it has a download means to download a device driver from management server equipment when the

judgment result of a version has a difference Only when version up is performed, the newest device driver can be downloaded to a peripheral device, and it can save for a preservation means. In each information processor, the newest device driver can be downloaded through an internal network, and the device driver in each information processor can be easily upgraded from the preservation means of a peripheral device. [0024] Moreover, when the difference with the version information of the device driver which the information processor which said notice means is as a result of [ by said version judging means ] a judgment in this invention, and is stored in said version storing means is using, and the version information of the device driver obtained from management server equipment by said inquiry means arises, it is characterized by to perform the inquiry of whether to perform install of the device driver downloaded with said download means to this information processor.

[0025] If this invention is followed and a difference will arise in the version information of the device driver which the information processor is using, and the version information acquired by the inquiry from management server equipment, it will be asked to an information processor whether a notice means performs install of the device driver downloaded with the download means. Since the user of each information processor can know that version up of a device driver is performed and the newest device driver is saved at the peripheral device, he can judge whether it installs if needed.

[0026] Moreover, said download means is characterized by to download each file which constitutes a device driver from the address corresponding to each file held by the address maintenance means, including further an address maintenance means hold the address of two or more management server equipments in which the file said whose device driver is constituted by multiple files and constitutes a device driver from this invention is stored corresponding to a file.

[0027] If this invention is followed, since multiple files will constitute the device driver and a file will be distributed and managed with two or more management server equipments, it becomes possible to mitigate the traffic to management server equipment, and the newest device driver accompanying version up can be distributed to a high speed.

[0028] Moreover, it is characterized by said inquiry means accessing this management server equipment by this invention using the log in password of the address of the management server equipment set up beforehand.

[0029] Since it will access using the log in password of the address of management server equipment in case a peripheral device performs an inquiry to management server equipment if this invention is followed, access from a peripheral device or an information processor which does not acquire the log in password can be prevented, and security can be raised.

[0030] Moreover, by this invention, said inquiry means is characterized by having a

password acquisition means to gain the log in password in the case of accessing next time from said management server equipment, and a password maintenance means to hold the log in password gained by the password acquisition means.

[0031] The log in password in the case of performing next access, after the inquiry means of a peripheral device accesses management server equipment if this invention is followed is gained from management server equipment by the password acquisition means, and what is held by the password maintenance means is used for it. By this, management server equipment becomes possible [ forming a password according to an individual for every peripheral device ], and can prevent unjust access effectively.

[0032] Moreover, an investigation program acquisition means by which said inquiry means acquires OS environmental research program from said management server equipment by this invention, An investigation program sending means to send OS environmental research program which answered the demand from said information processor and the investigation program acquisition means acquired to this information processor, A results-of-an-investigation acquisition means to gain the results of an investigation by this OS environmental research program from this information processor, A results-of-an-investigation sending means to send the results of an investigation gained by the results-of-an-investigation acquisition means to this management server equipment is included. Said download means It is characterized by downloading the device driver offered from this management server equipment corresponding to the results of an investigation sent by the results-of-an-investigation sending means.

[0033] If this invention is followed, the inquiry means of a peripheral device will send the investigation program which acquired OS environmental research program from management server equipment with the investigation program acquisition means, and was acquired with the investigation program sending means according to the demand from an information processor to an information processor. A results-of-an-investigation acquisition means gains the results of an investigation by OS environmental research program from an information processor. A results-of-an-investigation sending means sends the gained results of an investigation to management server equipment. A download means downloads the device driver offered from management server equipment corresponding to results of an investigation. It becomes upgradable [ the device driver which suited the hardware and software of the information processor connected through a network by this ]. Moreover, OS environmental research program can acquire OS environmental research program which can be used effectively, even when it cannot respond only by storing in the peripheral device beforehand that what is necessary is just to prepare OS environmental research program for the OS for management server equipment, even if OS which an information processor carries becomes new, since it is provided from management server equipment.

[0034] Moreover, said download means is characterized by downloading the device driver of the version chosen by the version selection means from management server equipment including a version information acquisition means to acquire the version information of the device driver of two or more versions by which said inquiry means is saved to said management server equipment by this invention, and a version selection means to choose the version of the device driver made to install in said information processor.

[0035] If this invention is followed, the inquiry means of a peripheral device will acquire the version information about the device driver of two or more versions saved to management server equipment with a version information acquisition means, and will choose the version of the device driver made to install in the information processor connected through a network by the version selection means. Since a download means downloads the device driver of the version chosen by the version selection means from management server equipment, it can download the device driver of the version which suited to the information processor. Generally, by version up of a device driver, a function is added, as a result, a program size also becomes large, and the need for storage capacity also increases. By choosing the version made to install according to the environment of an information processor, the correspondence to a wide range information processor is attained.

[0036] Moreover, by this invention, said preservation means is characterized by saving the device driver for two or more OS's.

[0037] With the information processor connected through a network to a peripheral device, if this invention is followed, since the device driver for two or more OS's is saved for the preservation means of a peripheral device, even if two or more kinds of OS's are used, a device driver can be offered from a peripheral device and the traffic to management server equipment can be mitigated.

[0038] Moreover, by this invention, said download means downloads the device driver to which cipher processing was performed from said management server equipment, and is characterized by having a decryption means to decode this code.

[0039] If this invention is followed, since a download means to download a device driver from management server equipment to a peripheral device downloads the device driver to which cipher processing was performed with management server equipment and a code is decoded with a decryption means, it will be guaranteed that the contents of the device driver are not changed during download. Since the download means in a peripheral device performs decode of a code, it is not necessary to decode a code and the device driver saved for the preservation means can be used as it is in the information processor connected through a network.

[0040] Moreover, by this invention, said inquiry means is characterized by having a file list receiving means to receive the file list corresponding to the configuration information of the device concerned by which a configuration information transmitting

means to transmit to said management server equipment, and the configuration information transmitting means transmits the configuration information of the peripheral device concerned from this management server equipment.

[0041] If this invention is followed, in case the inquiry means of a peripheral device will perform an inquiry to management server equipment, the configuration information of a peripheral device is transmitted with a configuration information transmitting means, and a file list receiving means receives the file list corresponding to the configuration information of the peripheral device concerned which is transmitted from management server equipment. Since management server equipment should just create a file list corresponding to the configuration information sent from the configuration information transmitting means of a peripheral device, it does not need to store a device driver beforehand according to the configuration of a peripheral device, and can save the storage capacity in management server equipment.

[0042] Moreover, by this invention, said inquiry means is characterized by notifying OS information on a device driver and the version information of this device driver which were installed in said host equipment at the time of the inquiry to said management server equipment to this management server equipment.

[0043] If this invention is followed, since OS information on the device driver installed in the information processor at the time of the inquiry to the management server equipment by the inquiry means of a peripheral device and the version information of a device driver will be notified, the information about OS with the high frequency currently used with the information processor etc. can be acquired, and OS which needs to further development of a device driver can be deduced.

[0044] Moreover, said download means is characterized by answering the detection result of an access-less detection means, and starting download of said device driver including an access-less detection means to detect that this invention does not have said information processor to predetermined time access.

[0045] If this invention is followed, since that a peripheral device starts download of a device driver from management server equipment performs that there is no predetermined time access from an information processor to a peripheral device based on the detected result with an access-less detection means, access to the peripheral device in an internal network environment can download a device driver in few condition, and can download a device driver at a high speed.

[0046] Moreover, by this invention, said inquiry means is characterized by asking the version information of a device driver using HTTP to said server equipment.

[0047] If this invention is followed, since the version information of a device driver is asked to management server equipment using HTTP, the inquiry means of a peripheral device can pass the security system introduced into the Gateway prepared between the network of not only the manager of an internal network but the interior, and the network to the exterior, and can perform the inquiry of version information to external

management server equipment.

[0048] Moreover, by this invention, said notice means is characterized by performing the demand and notice to said information processor using HTTP.

[0049] If this invention is followed, in case the demand and notice to the information processor by which the notice means is connected to the peripheral device through the network will be performed A demand and a notice can be performed also to the information processor connected through an internal network with the protocol same since HTTP is used as the protocol used in case an inquiry is performed to external management server equipment. Since all can be processed with the same protocol, it becomes possible to connect with an information processor and management server equipment seamlessly.

[0050] Furthermore, as for this invention, two or more information processors are connected to a peripheral device through an internal network. In the approach for managing the version of the device driver installed in each information processor in order to control this peripheral device This peripheral device is connected to the management server equipment which manages this device driver through the network to the exterior. From this peripheral device, the inquiry of the version information of the device driver currently held at this management server equipment is performed. It is the version control approach of the device driver characterized by notifying the version information acquired from this management server equipment to each information processor from this peripheral device.

[0051] If this invention is followed, when it is installed by the peripheral equipment maker etc. and a peripheral device notifies the version information of a device driver to the information processor connected to the inquiry and the peripheral device instead of the information processor connected through a network to the management server equipment with which the device driver of the newest version is always saved, the user of an information processor can judge whether the device driver which the information processor is using is the newest version.

[0052]

[Embodiment of the Invention] Hereafter, based on a drawing, the configuration and actuation of operation of this invention of each gestalt are explained. The same reference mark is given to the part corresponding to the part preceded and explained by the following explanation, and the overlapping explanation is omitted.

[0053] Drawing 1 shows the network-system configuration as the 1st gestalt of operation of this invention. The network system of this operation gestalt contains the host computer 1 which are two or more information processors fundamentally, the digital compound machine 2 which is the peripheral device which can be shared with two or more host computers 1, and the device driver management server computer 3 which is management server equipment which manages the device driver for using the digital compound machine 2 with each host computer 1. The digital compound machine

2 can access the device driver management server computer 3 through the Internet 5 as a network to the exterior connected through Gateway 4. The digital compound machine 2 and two or more host computers 1 are connected through the intranet 6 which is an internal network. Intranet 6 is a network established in the interior of one enterprise objects, such as LAN, and is defended to unjust access from the outside. The Internet 5 is wide opened to the exterior widely. The Internet 5 and intranet 6 communicate using the communications protocol called the same HTTP (Hyper TextTransfer Protocol). Therefore, a communication link in intranet 6 and the communication link linked to the Internet 5 through Gateway 4 can be performed similarly. However, the fire wall for securing the security in intranet 6 etc. is prepared in Gateway 4, and the security system is built so that unjust access may not be performed to intranet 6 from the exterior through the Internet 5.

[0054] The host computer 1 was equipped with OS11, and has incorporated the printer driver 12 and the scanner driver 13 for using the digital compound machine 2 as a device driver which constitutes a part of OS11. In case the user using a host computer 1 uses the e-mail application 14, the document preparation application 15, etc., he uses indirectly the printer driver 12 and the scanner driver 13 which are carried in OS11. A host computer 1 is equipped also with the WWW (World Wide Web) client 16 for communicating using the Internet 5, and the graphical display of a Web page can be made to perform using WWW browser 17 for perusal. It can transmit to the digital compound machine 2 through intranet 6, and the document drawn up using the e-mail application 14, the document preparation application 15, etc. and the received document can be printed by the function as a printer which the digital compound machine 2 has.

[0055] The digital compound machine 2 has a function as a printer, and an available function as a scanner from a host computer 1 through intranet 6. The digital compound machine 2 is independent and can also be equipped with the function as copy equipment or facsimile apparatus. The digital compound machine 2 is equipped with OS21, and actuation as a printer and a scanner is performed by the printer control section 23 and the scanner control section 24 which are controlled by OS21, respectively. In order that the digital compound machine 2 may communicate with a host computer 1 or the device driver management server computer 3 through intranet 6 or the Internet 5, the WWW server / client 22 is formed. The digital compound machine 2 is equipped with the version information transceiver section 25 and the time management section 26 as a function for the digital compound machine 2 to manage the version of device drivers, such as the printer driver 12 carried in the host computer 1, and the scanner driver 13. The version information transceiver section 25 communicates with a host computer 1 or the device driver management server computer 3 through a WWW server / client 22. The time management section 26 controls the transceiver timing which the version information transceiver section 25



performs.

[0056] The device driver management server computer 3 performed various control based on OS31, and is equipped with the WWW server 32 for communicating with the digital compound machine 2 or a host computer 1 through the Internet 5. The digital compound machine 2 and a host computer 1 can access the device driver version Research and Data Processing Department 33, and the printer driver file 34 and the scanner driver file 35 through the WWW server 32. The printer driver 12 and the scanner driver 13 which are carried in a host computer 1 are stored in the printer driver file 34 and the scanner driver file 35, respectively. When version up is performed to a printer driver 12 or the scanner driver 13, the new printer driver 12 and the scanner driver 13 which were upgraded are stored in the printer driver file 34 or the scanner driver file 35, respectively. The version information of the printer driver 12 stored in the printer driver file 34 and the scanner driver file 35 or the scanner driver 13 is stored in the device driver version Research and Data Processing Department 33. About the version by which the serious defect was discovered by the printer driver 12 or the scanner driver 13, it is deleted from the printer driver file 34 or the scanner driver file 35.

[0057] Drawing 2 shows roughly the procedure of the version control of the device driver in the operation gestalt of drawing 1. The device driver management server computer 3 is standing by at step a10 after starting actuation, according to the demand from the digital compound machine 2, is step a11, transmits the version information of the device driver stored in the device driver version Research and Data Processing Department 33 as management server equipment, and returns to standby of step a10 again. The digital compound machine 2 performs the standby for fixed preparation etc. at step a21 after initiation of operation, URL (Uniform Resource Locator) whose version information transceiver section 25 as an inquiry means it is step a22 and is the address of the device driver management server computer 3 is specified, and it accesses it, and requires version information. At step a23, if the version information of a device driver is received, by step a24, the version information transceiver section 25 as a notice means will transmit the version information of a device driver to each host computer 1, and will return to step a21. If it stands by at step a31 after initiation of operation and the version information of a device driver is transmitted from the digital compound machine 2, each host computer 1 will receive the version information of a device driver at step a32, and will return to step a31.

[0058] Therefore, shortly after the digital compound machine 2 always accessed the device driver management server computer 3, is demanding the version information of a device driver and receives version information, it is transmitted to each host computer 1. With each host computer 1, the information on the device driver of the latest version can always be acquired, and the user using a host computer 1 can judge easily whether the device driver currently used is the newest thing.

[0059] Moreover, since the digital compound machine 2 performs the inquiry of the version information of a device driver to the device driver management server computer 3 using HTTP, it can pass security systems, such as a fire wall introduced into Gateway 4 prepared between not only the network administrator of intranet 6 but the intranet 6, and the external Internet 5, and can perform the inquiry of version information to the external device driver management server computer 3. Moreover, since the demand and notice through the internal intranet 6 to a host computer 1 are also performed using HTTP, all can be processed with the same protocol, it is seamless and connection between a host computer 1 and the device driver management server computer 3 can be made.

[0060] Drawing 3 shows the procedure of the version control as the 2nd gestalt of operation of this invention. The system configuration of this operation gestalt is the same as that of drawing 1. The version information transceiver section 25 of the digital compound machine 2 of drawing 1 operates independently as the version information receive section as an inquiry means, and the version information transmitting section as a notice means. A version information receive section is step b11 after initiation of operation, and stands by fixed time amount. This fixed time amount is controlled by the time management section 26 which operates as a timer means. Fixed time amount is beforehand set as timing, such as daily fixed time of day, and a week as frequency, such as once. After standby of step b11 is completed, at step b12 and step b13, like each step of step a22 of drawing 2, and step a23, version information is required of the device driver management server computer 3, and version information is received. It stores in the memory in which the version information of the received device driver is prepared in the version information transceiver section 25 at step b14.

[0061] The actuation as the version information transmitting section stands by at step b21 after initiation of operation. At step b22, it judges whether the version information of a device driver is storing ending in the memory inside the version information transceiver section 25. When version information is judged not to be storing ending, it returns to step b21. At step b22, when the version information of a device driver is judged to be storing ending, by step b23, the version information of a device driver is transmitted to a host computer 1, and it returns to step b21.

[0062] With this operation gestalt, since access to the device driver management server computer 3 in step b12 is performed for every fixed time amount set up by the time management section 26, access can mitigate the traffic which does not occur frequently and minds the Internet 5.

[0063] Drawing 4 shows the procedure of the version control as the 3rd gestalt of operation of this invention. This operation gestalt also operates by the system configuration of drawing 1. The version information transceiver section 25 operates independently in a version information receive section and the version information

transmitting section like the operation gestalt of drawing 3 . Inquiry time amount is assigned to a version information receive section next time. If a version information receive section starts actuation, only inquiry time amount will stand by at step c11 the next time set up. After standby, at step c12, it accesses by specifying URL of the device driver management server computer 3, and version information and an updating predetermined time are required as an inquiry means and a spacing acquisition means, respectively. At step c13, the version information and the renewal predetermined time of version information of a device driver are received from the device driver management server computer 3. The version information of a device driver is stored in the memory with which the version information transceiver section 25 is equipped at step c14. As a spacing modification means, based on the renewal predetermined time of version information, the next inquiry time amount of the version control section 26 is set up, and it returns to step c11 at step c15.

[0064] Actuation of the version information transmitting section stands by at step c21 after initiation of operation, and the version information of a device driver judges whether it is storing ending in the memory in the version information transceiver section 25 at step c22. When it is judged that it is not storing ending, it returns to step c21. At step c22, when the version information of a device driver is judged to be storing ending, as a notice means, the version information of a device driver is transmitted to a host computer 1, and it returns to step c21 by step c23. Although actuation of this operation gestalt is similar to the operation gestalt shown in drawing 3 , when it understands the schedule of version up of a device driver beforehand, the digital compound machine 2 is different from the device driver management server computer 3 in that the next updating predetermined time is required with version information. From the device driver management server computer 3, reception of the next renewal predetermined time of version information changes a setup of the next inquiry time amount of a version information receive section. Therefore, it generates only in necessary minimum but access to the device driver management server computer 3 can mitigate further the traffic by the communication link through the Internet 5.

[0065] Drawing 5 shows the operations sequence of the version control as the 4th gestalt of operation of this invention. This operation gestalt also operates by the system configuration of drawing 1 . Although this operation gestalt is similar to the operation gestalt of drawing 2 , the point of being the timing to which the digital compound machine 2 accesses the device driver management server computer 3 at the detection time of power-source ON is different. If detect stand going up of a power source, it starts, a detection means is established in the version information transceiver section 25 and power-source ON is detected, at step d11, URL of a device driver management server computer will be specified, and version information will be required. If the version information of a device driver is transmitted from the

device driver management server computer 3, it will receive at step d12. At step d13, the version information of a device driver is transmitted to a host computer 1. Each step from the above step c11 to step c13 is fundamentally [ as step a22 to the step a24 of drawing 2 ] equivalent.

[0066] This operation gestalt performs access to the external device driver management server computer 3 at the time of power-source ON. Since the internal intranet 6 is also used in case the digital compound machine 2 accesses the device driver management server computer 3 through the external Internet 5, the use to the digital compound machine 2 from the host computer 1 connected through intranet 6 receives a limit. There is little possibility of a power source that will start and the digital compound machine 2 will be used from a host computer 1 at the time, and it can acquire version information, without reducing the effectiveness as a network system. By being able to perform detection of power-source ON with a comparatively easy configuration, and preparing for the version information transceiver section 25, the need of performing time management in the version control section 26 of drawing 1 etc. is abolished, and if the time management section 26 is omitted, a cost cut can be aimed at.

[0067] Drawing 6 shows the operations sequence of the version control by the digital compound machine 2 as the 5th gestalt of operation of this invention. This operation gestalt acquires the version of a device driver from the device driver management server computer 3 through step a21, step a22 and step a23 and the respectively same step e11, step e12, and step e13 of an operation gestalt of drawing 2. The point which is different from the operation gestalt of drawing 2 with this operation gestalt is a point that a version information storing means to store the version information of the device driver currently used for the version information transceiver section 25 with the present host computer 1 is established. Moreover, at step e14, it judges whether the version information of the device driver which received as a version judging means is the same as the version which is carrying out current maintenance. If the version is the same, it will return to step e11. At step e14, when judged with version information not being the same, as a notice control means, the version information of the device driver received from the device driver management server computer 3 is transmitted to a host computer 1, and it controls by step e15 to update the version information of a device driver at step e16, and returns to step e11 by it.

[0068] With this operation gestalt, the version information acquired from the device driver management server computer 3 When in agreement with the version information stored in the version information transceiver section 25 (i.e., when the version of a device driver is not updated) When transmission to the host computer 1 in step e15 is not performed and version information is not in agreement (i.e., only when the device driver is updated), while transmitting at step e15 He is trying to newly store the updated version information at step e16. Therefore, the traffic in user network

environments, such as the internal intranet 6, can be reduced.

[0069] Drawing 7 shows the operations sequence of the system of drawing 1 as the 6th gestalt of operation of this invention. Only when this operation gestalt is similar to the operation gestalt shown in drawing 2 and a demand of version information is given to the digital compound machine 2 from a host computer 1, the digital compound machine 2 is requiring version information from the device driver management server computer 3, and transmitting the received version information only to the host computer 1 with which the demand of version information was performed.

[0070] That is, actuation of the device driver management server computer 3 from step f11 to step f12 is equivalent to each step of step a10 of drawing 2, and step a11. Actuation of the digital compound machine 2 is step f21 after initiation of operation, and stands by. Each step of step f22 after standby termination, step f23, and step f24 is equivalent to each step of step a22 of drawing 2, step a23, and step a24 respectively.

[0071] A host computer 1 requires version information from the digital compound machine 2 after standby of step f31 at step f32. As a version control means, a demand of version information is received from a host computer 1 waiting, it is [ following ] from step f22 to step f24, and the version information transceiver section 25 of the digital compound machine 2 transmits the version information of step f21 acquired from the device driver management server computer 3 to a host computer. A host computer 1 is step f33, receives the version information of the device driver transmitted through intranet 6 from the digital compound machine 2, and returns to step f31.

[0072] This operation gestalt transmits version information only to the host computer 1 which requires transmission of version information [ the configuration which transmits version information to many and unspecified host computers 1 connected to intranet 6 ] in the case of new installation of an application program, as shown in drawing 2. For example, the host computer 1 which the present version is enough as and does not have the need for version up can be excepted from the candidate for transmitting of the version information acquired from the device driver management server computer 3 since version information is not required, and the traffic of the intranet 6 which is a user network environment can be reduced.

[0073] Drawing 8 shows the configuration of the computer network system as the 7th gestalt of operation of this invention. With this operation gestalt, a host computer 1, Gateway 4, the Internet 5, and intranet 6 are equivalent to the operation gestalt of drawing 1 respectively. The point which is different from the operation gestalt of drawing 1 with this operation gestalt is that the driver transceiver section 47 which transmits and receives a device driver, and the driver file library 48 as a preservation means to save a device driver are established in the interior in the digital compound machine 42. In order to transmit and receive a device driver through intranet 6 or the

Internet 5, transmission and reception of a device driver are possible also for OS51, or the WWW server / client 52, and it is made. The device driver which the digital compound machine 42 holds is acquired from the device driver management server computer 53. The device driver transmitting section 56 which transmits not only the version information of a device driver but the device driver itself to peripheral devices, such as the digital compound machine 42, is formed in the device driver management server computer 53. OS61 and the WWW server 62 are also constituted so that transmission of the device driver to the digital compound machine 42 may be possible.

[0074] Drawing 9 shows the procedure of the version control actuation by the system configuration of drawing 8. Actuation of the digital compound machine 42 in this operation gestalt is similar to actuation of the digital compound machine 2 in the operation gestalt shown in drawing 6. That is, if the device driver management server computer 53 stands by at step g11 and has a demand of the version information from the digital compound machine 42, it will repeat the actuation which transmits version information of a device driver at step g12. The digital compound machine 42 operates step g21 to each step of step g22, step g23, and step g24 equally, respectively with each step of step e11 of the operation gestalt of drawing 6, step e12, step e13, and step e14. Step g25 of this operation gestalt requires download of the device driver of the version which updated the version information of the device driver which received from the device driver management server computer 53, and was updated to the device driver management server computer 53 at step g26. At step g27, the device driver transmitted from the device driver management server computer 53 is received, and it returns to step g21. That is, at step g26 and step g27, a device driver is downloaded as a download means. By device driver management server computer 53, the device driver stored as the printer driver file 34 or a scanner driver file 35 at step g32 according to the download demand of a device driver from the digital compound machine 42 is transmitted to the digital compound machine 2 after standby of step g31, and it returns to step g31.

[0075] With this operation gestalt, the newest device driver will always be stored in the digital compound machine 42. Each host computer 1 can download the newest device driver through intranet 6 at any time from the digital compound machine 2, and can upgrade a device driver easily. Since each host computer 1 does not need to access the direct device driver management server computer 53 through the Internet 5 for version up of a device driver, the traffic to the device driver management server computer 53 can be decreased.

[0076] Drawing 10 shows the version control procedure of the network system of drawing 8 as the 8th gestalt of operation of this invention. This operation gestalt is similar to the operation gestalt shown in drawing 9, and performs the comparison of the version information in step h11 at step g24 respectively like a procedure [ version information / after actuation of each step from step g21 of drawing 9 to step g23 ].

Actuation of each step of the step h12 after version information was judged not to be the same as the version held now at step h11, step h13, and step h14 is equally performed, respectively with each step of step g25 of drawing 9 , step g26, and step g27. In this operation gestalt, the digital compound machine 42 performs the inquiry about the install demand of whether to install the upgraded device driver to each host computer 1 as a notice means at step h15 next. The user of a host computer 1 who receives this inquiry can recognize that the newest driver exists.

[0077] Drawing 11 shows the operations sequence of the version control in the system configuration of drawing 8 as the 9th gestalt of operation of this invention. With this operation gestalt, the log in password to the address of the device driver management server computer 53 beforehand set as the inquiry of the version information over the device driver management server computer 53 from the version information transceiver section 25 as an inquiry means of the digital compound machine 42 is used. That is, the device driver management server computer 53 is step i11 after initiation of operation, and stands by and receives a password. If a password is received, a password will be judged at step i12 and a judgment result will be transmitted. At step i13, it is as a result of [ of step i12 ] a judgment, and when a password is not right, it returns to step i11. At step i13, a password transmits the version information of the device driver stored in the device driver management server computer 53 at step i14 at the time of the right, and returns to step i11. With the digital compound machine 42, it stands by at step i21 after initiation of operation, and at step i22, URL of the device driver management server computer 53 is specified, and a password is transmitted. The log in password set up beforehand is used for this password corresponding to URL of the device driver management server computer 53. At step i23, the password judging result from the device driver management server computer 53 is received. At step i24, when a judgment result is not O.K., it returns to standby of step i21. At step i24, when a judgment result is O.K., step i25 receives the version information of the device driver transmitted from the device driver management server computer 53.

[0078] Only the peripheral device which has the log in password set up beforehand by the device driver management server computer 53 side becomes accessible, and it becomes impossible to access many and unspecified host computers 1 which are not connected to such a peripheral device with this operation gestalt at the device driver management server computer 53. The security to device driver management can be raised by this.

[0079] Drawing 12 shows actuation of the version control in the system configuration of drawing 8 as the 10th gestalt of operation of this invention. This operation gestalt is similar to the operation gestalt shown in drawing 11 , and after accessing difference using the log in password set up beforehand, it is setting up anew the log in password in the case of accessing next time with version information. That is, each step of step

j11 of drawing 12 , step j12, and step j13 performs actuation respectively equivalent to each step of step i11 of drawing 11 , step i12, and step i13. The device driver management server computer 53 is step j13, when a password is equal, transmits the version information of the device driver stored in the device driver management server computer 53, and a next password as step j14, and returns to step j11. The actuation by the side of the digital compound machine 52 has respectively actuation of each step of step j21, step j22, step j23, and step j24 equivalent to each step of step i21 of drawing 11 , step i22, step i23, and step i24. The digital compound machine 42 receives a password as a password acquisition means at step j25 the version information of the device driver transmitted from the device driver management server computer 53, and next time, when a judgment result is O.K. at step j24. Next, at step j26, as a password maintenance means, a password is stored next time and it holds in preparation for a next inquiry.

[0080] With this operation gestalt, if the device driver management server computer 53 can also specify the password which changes individually and serially every digital compound machine 42 and such a password is specified, it will become accessible [ the digital compound machine 42 ] only with the password. The device driver management server computer 53 can prevent unjust access, even when access with the same password can be detected, it can also be judged as unjust access and a password is known for a certain cause by the third person.

[0081] Drawing 13 shows version control actuation of the network system of drawing 8 as the 11th gestalt of operation of this invention. However, the file which a device driver divides into multiple files, consists of these operation gestalten, and constitutes a device driver shall be distributed and prepared for the device driver management server computer 53 currently installed two or more sets. In response to this, the version information transceiver section 25 of the digital compound machine 42 holds the address of two or more device driver management server computers 53 as an address maintenance means. As a download means, in case a device driver is downloaded, sequential access is carried out from the first device driver management server computer 53, and sequential download of the file which constitutes a device driver is carried out. Therefore, the device driver management server computer 53 can be distributed, it can mitigate that access for download concentrates on the specific device driver management server computer 53, and traffic becomes heavy at the time of version up etc., and the device driver through the network accompanying version up of a device driver can be distributed to a high speed.

[0082] Drawing 13 shows the procedure in which the digital compound machine 42 downloads the upgraded device driver. At step k11, the version information transceiver section 25 of the digital compound machine 42 gains the number of the device driver management server computers 53 which divided into multiple files the device driver set as the object of download, and have managed it. Next, the address of



the device driver management server computer 53 which has managed the file of the beginning of the files which constitute a device driver from a step k12 is gained. The address of the device driver management server computer 53 is beforehand specified by the manufacture manufacturer of the digital compound machine 42 etc. Next, at step k13, the device driver file currently held by the device driver management server computer 53 is downloaded from the device driver management server computer 53 of the address specified. At step k14, the address of the device driver management server computer 53 which has managed the next file is gained. At step k15, the device driver management server computer 53 which downloaded the file reaches to the number of the device driver management server computers 53 gained at step k11, and it judges whether it is the last device driver management server computer 53. When it is judged that it is not last, it returns to step k13 and download of a file is continued.

[0083] The address of the following device driver management server computer 53 gained at step k14 can be beforehand specified by the manufacture manufacturer of a peripheral device etc. like the number of the device driver management server computers 53 gained at step k11, and the address of the first device driver management server computer 53 gained at step k12. Moreover, whether a device driver is divided into multiple files actually accesses the device driver management server computer 53, and you may make it become clear. That is, the manufacture manufacturer of a peripheral device etc. specifies only one device driver management server computer 53, and may be made to become clear by the result of having accessed the device driver management server computer 53 from the user side, and having downloaded the device driver.

[0084] A device driver is divided into multiple files, when it is necessary to download dispersedly from two or more device driver management server computers 53, the accessed device driver management server computer 53 turns into the first device driver management server computer 53, and a peripheral device can be told about the number of the device driver management server computers 53 holding a file required to constitute as a device driver, or its address from the first device driver management server computer 53. Furthermore, in case two or more device driver management server computers 53 download a file from the device driver management server computer 53 to precede in order of the file which constitutes a device driver, they can specify the address of the consecutive device driver management server computer 53. The last device driver management server computer 53 can tell that it is the last device driver management server computer 53 by not specifying the address of the consecutive device driver management server computer 53.

[0085] Drawing 14 shows the procedure in which the network system shown in drawing 8 performs version control of a device driver as the 12th gestalt of operation of this invention. With this operation gestalt, the program which investigates the OS environment of a host computer 1 is stored in the device driver management server

computer 53, and according to the demand from a host computer 1, the digital compound machine 42 can require the OS environmental research program of the device driver management server computer 53, and can be gained to it. If the digital compound machine 42 acquires OS environmental research program, it transmits to a host computer 1, and the OS environment will be made to investigate, results of an investigation will be received, and it will transmit to the device driver management server computer 53. From the device driver management server computer 53, the device driver corresponding to the results of an investigation of OS environmental research program is downloadable.

[0086] That is, as shown in drawing 14 , if the device driver management server computer 53 stands by at step I11 and has the demand from the digital compound machine 42, it will transmit OS environmental research program at step I12, and will return to standby of step I11 again. If it relates to transmission of a device driver, it stands by at step I21, if the results of an investigation of OS environmental research program are transmitted, it will receive, and according to the result, a device driver is transmitted at step I22, and it returns to standby of step I21.

[0087] In the digital compound machine 42, it operates as an investigation program acquisition means according to a demand of OS investigation environmental program from a host computer 1 following standby at step I31. That is, OS environmental research program is required of the device driver management server computer 53 at step I32, and at step I33, as an investigation program sending means, OS environmental research program is received from the device driver management server computer 53, and it transmits to a host computer 1. At step I34, the results of an investigation from a host computer 1 are received as a results-of-an-investigation acquisition means, it transmits to the device driver management server computer 53 as a results-of-an-investigation sending means, the device driver corresponding to results of an investigation is downloaded from the device driver management server computer 53 as a download means, and it returns to step I31. A host computer 1 is step I42 in step I41 after standby, and gives a demand of OS environmental research program to the digital compound machine 42. At step I43, OS environmental research program is received from the digital compound machine 42, the results of an investigation which performed OS environmental research program at step I44 are transmitted to the digital compound machine 42, and it returns to step I41.

[0088] With this operation gestalt, it becomes upgradable according to the device driver to which it suited the hardware and software of a host computer 1 since the device driver corresponding to the results of an investigation of OS environmental research program to a host computer 1 was stored in the device driver file library 48 of the digital compound machine 42 and downloaded and installed in the host computer 1 through the Internet 6 at any time. Moreover, OS environmental research program can be managed not by the digital compound machine 42 which is a peripheral

device but by device driver management server computer 53, and can be acquired from the device driver management server computer 53. By this, if OS environmental research program which suited the OS is prepared for the device driver management server computer 53 by device driver management server computer 53 even if it is a case so that OS11 of a host computer 1 may be completely changed into other OS's, OS environmental research program in which environmental research sufficient also with the host computer 1 which carried new OS is possible can be prepared.

[0089] Drawing 15 shows the procedure of the version control by the network system of drawing 8 as the 13th gestalt of operation of this invention. With this operation gestalt, the device driver of two or more versions is stored in the device driver management server computer 53, and as a version information acquisition means, the version information transceiver section 25 of the digital compound machine 42 acquires the version information of all classes, transmits to a host computer 1, and downloads only the device driver demanded from the host computer 1. That is, at step m11, an inquiry is performed to the device driver management server computer 53, and all the version information of the device driver stored is acquired. At step m12, the information on all the versions of the gained device driver is transmitted to a host computer 1. At step m13, the device driver corresponding to the version information as which install was required is chosen from a host computer 1 as a version selection means, and it downloads from the device driver management server computer 53 as a download means.

[0090] As for a device driver, it is common that a function is added by version up. However, the added function may not necessarily be required on the environment of the hardware in a host computer 1, or software. Since a program size also becomes large, many device drivers to which the function was added will be needed by addition also according [ the memory of a host computer 1 ] to version up. For this reason, according to two or more demands, by some versions, version up can also be performed so that the addition of a function may be suppressed and a program size may be thought as important. When two or more versions exist, it is possible to choose the version made to install with a host computer 1; it is making, and correspondence to the wide range host computer 1 can be enabled.

[0091] Drawing 16 shows the operations sequence of the version control in the network system of drawing 8 as the 14th gestalt of operation of this invention. It is this operation gestalt's being similar to the operation gestalt of drawing 15 , and requiring OS information which all the device drivers that difference's is the digital compound machine 42 and are stored in the device driver management server computer 53 support, respectively, downloading all those device drivers beforehand and saving them.

[0092] That is, at step n11 of drawing 16 , the digital compound machine 42 acquires all OS information that the device driver stored in the device driver management

server computer 53 supports. At step n12, a device driver is downloaded for every OS information. It judges whether at step n13, download of the device driver corresponding to all OS information is completed. When having not ended, it returns to step n12. With this operation gestalt, since the device driver corresponding to OS currently used with all the host computers 1 connected to the digital compound machine 42 through the ITORA network 6 is downloaded and saved beforehand at the digital compound machine 42, compared with the case where only the device driver which the need produced is downloaded serially, the traffic by access to the device driver management server computer 53 is mitigable.

[0093] Drawing 17 shows the operations sequence of the version control in the network system of drawing 8 as the 15th gestalt of operation of this invention. This operation gestalt is an operation gestalt shown in drawing 9, and is similar to the part of the actuation which downloads a device driver from the device driver management server computer 3 to the digital compound machine 2. That is, the device driver management server computer 3 stores in the printer driver file 34 or the scanner driver file 35 the device driver by which cipher processing was carried out. If there is a demand from after standby and the digital compound machines 2 and 42 in step o11, at step o12, the device driver file which is stored and by which cipher processing was carried out will be transmitted, and it will return to step o11. In the digital compound machines 2 and 42, the demand of the download of a device driver by which cipher processing was carried out is given to the device driver management server computers 3 and 53 at step o21. At step o22, the device driver by which cipher processing was carried out from the device driver management server computers 3 and 53 is received. At step o23, a code is decoded as a decryption means in the digital compound machine 53, and a device driver is stored in the driver file library 48 in the condition.

[0094] In encryption processing, since it turns out whether the error has arisen as data even if the contents of data are unknown, it can judge easily whether the contents of the device driver by which cipher processing was carried out are changed into un-wanting by the third person. When the driver transceiver section 47 of the digital compound machine 42 is equipped with the function of decryption, that a third person does not change into un-wanting decodes the guaranteed device driver and it stores in the driver file library 48, with a host computer 1, the decoded device driver can be downloaded and used from the driver file library 48, and the program which decodes a code is not needed.

[0095] Drawing 18 shows the operations sequence of version up with the network system shown in drawing 8 as the 16th gestalt of operation of this invention. With this operation gestalt, it classifies for every configuration of peripheral devices, such as a printer driver 12 and the scanner driver 13, and it divides into the device driver management server computer 53 like the printer driver file 34 or the scanner driver

file 35, and is stored in it. A peripheral device transmits the information which shows the configuration to the device driver management server computer 53, the device driver management server computer 53 transmits the file list corresponding to the configuration information, and a peripheral device downloads a required device driver one by one along with the file list.

[0096] That is, the version information transceiver section 25 as an inquiry means transmits the configuration information which shows that the printer and the scanner are contained to the digital compound machine 42, for example to the device driver management server computer 53 at step p11 as a configuration information transmitting means. At step p12, the file list corresponding to a configuration is received from the device driver management server computer 53 as a file list receiving means. At step p13, all the device driver files corresponding to a configuration are downloaded along with a file list.

[0097] With this operation gestalt, the device driver management server computer 53 stores it for every configuration of a peripheral device rather than stores a device driver for every peripheral device. For example, the device driver of a printer part is common to a copying machine and a printer, and the device driver of a scanner part is further needed for a copying machine by them. If the device driver is stored for every peripheral device as it said that the device driver management server computer 53 was equipped with the device driver of a printer part and a scanner part to a copying machine, and was equipped with the device driver of only a printer part to a printer, it will overlap like a printer part and it will be necessary to store. Like [ as a printer part and a scanner part ], if the device driver is stored for every configuration of a peripheral device, the device driver which overlapped the device driver management server computer 53 is no longer stored, and can reduce the amount of data required for storing sharply. By device driver management server computer 53, since the device driver is stored in the hard disk etc., saving of a hard disk etc. can be aimed at by storing a device driver for every configuration of every peripheral device and a peripheral device.

[0098] Drawing 19 shows the management procedure of the device driver in the network system of drawing 8 as the 17th gestalt of operation of this invention. It is that this operation gestalt is similar to the operation gestalt shown in drawing 15, the version information transceiver section 25 of the digital compound machine 42 requires and acquires OS information which all the device drivers stored in the device driver management server computer 53 support, respectively, and difference transmits it to a host computer 1 by making it into installable OS information as an inquiry means. It is the point which downloads only the device driver which had an install demand from the host computer 1 from the device driver management server computer 53 to the digital compound machine 42. That is, at step q11, OS information corresponding to the device driver stored in the device driver management server

computer 53 is acquired. At step q12, the device driver is managed from the digital compound machine 42 by device driver management server computer 53 to the host computer 1, and installable OS information is notified. A host computer 1 judges whether install of a device driver is required with reference to OS information notified from the digital compound machine 42. If install of a device driver is required from a host computer 1, the digital compound machine 42 will download the device driver corresponding to OS information by which the install demand was carried out from the device driver management server computer 53 at step q13.

[0099] With this operation gestalt, with the host computer 1 which can grasp OS information and version information corresponding to the device driver installed in a host computer 1, and is connected through the Internet 5 by the device driver management server computer 53 side, since the information about OS with high operating frequency etc. can also be acquired, operating frequency is high, and OS which needs to further development preponderantly can also be deduced.

[0100] Drawing 20 shows the management activities of the device driver by the network system of drawing 8 as the 18th gestalt of operation of this invention. This operation gestalt is similar to the operation gestalt shown in drawing 9, and difference is performing download of a device driver to the digital compound machine 42, when there is no access from a host computer 1. That is, actuation to step r11 is performed on a par with step g21 of drawing 9, step g22, step g23, and step g24, and it stands by until access from a host computer 1 becomes that there is nothing at step r12 in predetermined time. If there is no access even if it carries out predetermined time progress, it will operate as an access-less detection means, and standby will be terminated. Actuation respectively equivalent to each step of step g25 of drawing 9, step g26, and step g27 is performed at each step to step r13, step r14, and step r15. Actuation of each step of step r21 by the side of the device driver management server computer 53 and step r22 is the same as actuation of each step of step g31 of drawing 9, and step g32.

[0101] With this operation gestalt, even if the version information of the device driver which received from the device driver management server computer 53 is judged to differ from the version which is carrying out current maintenance, after it does not generate the demand of download immediately, but standing by until access from each host computer 1 stops covering predetermined time and losing access at step r11, from the renewal of version information to download of a device driver is performed. Therefore, the traffic of user network environments, such as intranet 6, can start download of a device driver in few condition, and can download a device driver at high speed.

[0102] Each operation gestalt explained above can also be combined and used except for the part from which a fundamental view differs. Moreover, although the case where the digital compound machines 2 and 42 are used is explained as a peripheral device,

an independent printer, an independent scanner, etc. can also be used as a peripheral device.

[0103]

[Effect of the Invention] Since the peripheral device used by two or more information processors through a network notifies to the information processor by which the version information which can acquire version information from an inquiry and management server equipment is connected to the management server equipment to which the device driver for using a peripheral device is connected through an external network through an internal network with each information processor as mentioned above according to this invention, the user of an information processor can judge easily whether the device driver installed now is the newest version.

[0104] Moreover, according to this invention, it can supervise whether a peripheral device is the version of the newest [ device driver / by which the version information of a device driver is installed in the inquiry and the user of each information processor with the fixed time interval to management server equipment through the external network ].

[0105] Moreover, since the inquiry means of a peripheral device performs the inquiry of the version information of a device driver to management server equipment based on spacing which should carry out the inquiry of the version information acquired from a management server equipment side according to this invention, the inquiry doubled with the schedule of version up by the management server equipment side can be performed, a useless inquiry can be avoided, and the traffic to management server equipment can be mitigated.

[0106] Moreover, according to this invention, the delay based on being during the inquiry of version information is avoidable to the demand of use of the peripheral device from the information processor connected to it through an internal network since the inquiry means of a peripheral device can perform the inquiry of the version information over management server equipment to timing with little possibility in the warming up after powering on of being used, through an external network. Since a power source can start and detection can be performed comparatively easily, as compared with a configuration which is asked with a fixed time interval, a cost cut also becomes possible.

[0107] Moreover, since the notice to an information processor is performed only when there is a difference with the version information of the device driver which manages the version information of the device driver which the information processor connected to a peripheral device through an internal network is using with the peripheral device, and is saved to management server equipment according to this invention, the traffic in an internal network environment is mitigable.

[0108] Moreover, according to this invention, since a peripheral device makes only the information processor which is demanding the notice of version information among the

information processors connected through an internal network notify version information, it can reduce the traffic in an internal network environment.

[0109] Moreover, since a peripheral device downloads and saves a device driver from management server equipment according to this invention, the information processor connected through an internal network can be accessed to management server equipment, does not need to download a device driver, and can acquire a required device driver from a peripheral device through an internal network. For the preservation means of a peripheral device, since a device driver is downloaded from management server equipment based on the comparison of version information only when a difference arises in version information, the newest device driver can always be saved.

[0110] Moreover, according to this invention, since it asks whether the notice means of a peripheral device performs install of the downloaded device driver to the information processor connected through an internal network, the user of an information processor can recognize the newest driver downloading to a peripheral device and existing in it, and can judge whether the device driver of the newest version is installed.

[0111] Moreover, since according to this invention a device driver is divided into multiple files and it manages with two or more management server equipments, in case a device driver is downloaded to a peripheral device, a device driver can be divided and downloaded from two or more distributed management server equipments. At the time of version up of a device driver, since it is easy to concentrate access on management server equipment, by distributing and managing a device driver with two or more management server equipments, concentration of access to specific management server equipment can be avoided, and mitigation of traffic can be aimed at.

[0112] Moreover, since the log in password beforehand set up by the management server equipment side is used in case the inquiry means of a peripheral device accesses the management server equipment connected to an external network according to this invention, only the peripheral device which has a log in password can become accessible, and security can be raised.

[0113] Moreover, since the log in password used in case the inquiry means of a peripheral device accesses management server equipment can be specified from management server equipment according to this invention, access which uses the same password continuously can be detected easily, and unjust access can be prevented effectively.

[0114] Moreover, since the results of an investigation which gained the results of an investigation of the OS environment about an information processor where a peripheral device is connected through an internal network, and were gained are sent to management server equipment at OS environmental-research program which is



offered from management server equipment according to this invention, it becomes possible to acquire the device driver based on the results of an investigation of OS environmental-research program, and it becomes upgradable from management server equipment at the device driver which suited the operating environment of each information processor. Moreover, even when it carries new OS in an information processor, it can be made to correspond easily, since OS environmental research program is offered from management server equipment.

[0115] Moreover, when according to this invention a device driver upgrades and the device driver of two or more versions exists, a peripheral device can make the device driver of the version which suited the system of each information processor choose and install to two or more information processors connected through an internal network, and becomes possible [ corresponding to a wide range information processor ].

[0116] Moreover, according to this invention, since the device driver for two or more OS's is stored in the preservation means of a peripheral device, even if the information processor connected through an internal network to a peripheral device is using two or more OS's, a required device driver can be offered from a peripheral device. To extensive OS, a device driver can be managed only in an internal network, the inquiry to the management server equipment through an external network and the count of the download from management server equipment can be reduced, and the traffic of management server equipment can be mitigated.

[0117] Moreover, since according to this invention the device driver enciphered from management server equipment is downloaded and a code is decoded within a peripheral device, it is guaranteed that the downloaded contents of the device driver are not changed. In the information processor connected through an internal network, as it is not necessary to decode a code, the device driver of the newest version can also be transmitted and installed in each information processor certainly and quickly.

[0118] Moreover, according to this invention, since a device driver is obtained from management server equipment as a file list based on the configuration information of a peripheral device, by the management server equipment side, that what is necessary is just to create a file list based on the configuration information transmitted from a peripheral device, it becomes unnecessary to store the device driver according to the configuration of a peripheral device beforehand, and reduction of the storage capacity in management server equipment can be aimed at.

[0119] Moreover, according to this invention, since OS information and version information of a device driver which are installed in the information processor connected through an internal network are notified in case an inquiry is performed to management server equipment, by the management server equipment side, an inquiry means can know OS with high operating frequency with an information processor, and can specify OS which needs to further development of a device driver preponderantly.

[0120] Moreover, since download of a device driver is started when it detects that there is no access from two or more information processors connected through an internal network predetermined time according to this invention, in few condition, use of the peripheral device through the network by the user can download the device driver from the management server equipment through an external network, and can download a device driver at high speed.

[0121] Moreover, according to this invention, since the inquiry of the version information of a device driver is performed to management server equipment, using HTTP as a protocol, the inquiry means of a peripheral device can pass security systems, such as a fire wall introduced into the Gateway prepared between the network of not only the network administrator about an internal network but the interior, and an external network, and can perform the inquiry of version information from an external network to management server equipment.

[0122] Moreover, since the demand and notice which are performed with the information processor connected through an internal network from a peripheral device are performed according to this invention, using HTTP as a protocol, the inquiry to the management server equipment through an external network and the notice to the information processor through an internal network can be processed with the same protocol, and it becomes possible to connect management server equipment and an information processor seamlessly to a peripheral device.

[0123] Furthermore, according to this invention, it is installed by the peripheral equipment maker etc., and to the management server equipment with which the device driver of the newest version is always saved, a peripheral device is replaced with the information processor connected through an internal network, and performs an inquiry for the version information of a device driver through an external network. The user of an information processor can judge whether the device driver which the information processor is using is a thing in the newest version by notifying the result of an inquiry to an information processor through an internal network.

---

## DESCRIPTION OF DRAWINGS

---

[Brief Description of the Drawings]

[Drawing 1] It is the block diagram showing the rough system configuration as the 1st gestalt of operation of this invention.

[Drawing 2] It is the flow chart which shows the version control procedure of the device driver in the operation gestalt of drawing 1 .

[Drawing 3] It is the flow chart which shows the version control procedure of the device driver as the 2nd gestalt of operation of this invention.

[Drawing 4] It is the flow chart which shows the version control procedure of the device driver as the 3rd gestalt of operation of this invention.

[Drawing 5] It is the flow chart which shows the version control procedure of the device driver as the 4th gestalt of operation of this invention.

[Drawing 6] It is the flow chart which shows the version control procedure of the device driver as the 5th gestalt of operation of this invention.

[Drawing 7] It is the flow chart which shows the version control procedure of the device driver as the 6th gestalt of operation of this invention.

[Drawing 8] It is the block diagram showing the system configuration as the 7th gestalt of operation of this invention.

[Drawing 9] It is the flow chart which shows the version control procedure of the device driver in the operation gestalt of drawing 8 .

[Drawing 10] It is the flow chart which shows the version control procedure of the device driver as the 8th gestalt of operation of this invention.

[Drawing 11] It is the flow chart which shows the version control procedure of the device driver as the 9th gestalt of operation of this invention.

[Drawing 12] It is the flow chart which shows the version control procedure of the device driver as the 10th gestalt of operation of this invention.

[Drawing 13] It is the flow chart which shows the version control procedure of the device driver as the 11th gestalt of operation of this invention.

[Drawing 14] It is the flow chart which shows the version control procedure of the device driver as the 12th gestalt of operation of this invention.

[Drawing 15] It is the flow chart which shows the version control procedure of the device driver as the 13th gestalt of operation of this invention.

[Drawing 16] It is the flow chart which shows the version control procedure of the device driver as the 14th gestalt of operation of this invention.

[Drawing 17] It is the flow chart which shows the version control procedure of the device driver as the 15th gestalt of operation of this invention.

[Drawing 18] It is the flow chart which shows the version control procedure of the device driver as the 16th gestalt of operation of this invention.

[Drawing 19] It is the flow chart which shows the version control procedure of the device driver as the 17th gestalt of operation of this invention.

[Drawing 20] It is the flow chart which shows the version control procedure of the device driver as the 18th gestalt of operation of this invention.

[Description of Notations]

1 Host Computer

2 42 Digital compound machine

3 53 Device driver management server computer

4 Gateway

5 Internet

6 Intranet

11,21,31,51,61 OS

12 Printer Driver

13 Scanner Driver

14 E-mail Application

15 Document Preparation Application

16 WWW Client

17 WWW Browser

22 52 A WWW server / client

23 Printer Control Section

24 Scanner Control Section

25 Version Information Transceiver Section

26 Time Management Section

32 62 WWW server

33 Device Driver Version Research and Data Processing Department

34 Printer Driver File

35 Scanner Driver File

47 Driver Transceiver Section

48 Driver File Library

(19)日本国特許庁 (J P)

## (12) 公開特許公報 (A)

(11)特許出願公開番号

特開2001-125773

(P2001-125773A)

(43)公開日 平成13年5月11日(2001.5.11)

(51)Int.Cl. <sup>7</sup>	識別記号	F I	テマコード <sup>*</sup> (参考)
G 0 6 F 9/06	4 1 0	G 0 6 F 9/06	4 1 0 C 5 B 0 1 4
			4 1 0 Q 5 B 0 7 6
13/00	3 5 1	13/00	3 5 1 H 5 B 0 8 9
13/10	3 1 0	13/10	3 1 0 D
13/14	3 3 0	13/14	3 3 0 B
審査請求 未請求 請求項の数21 O L (全 24 頁)			

(21)出願番号 特願平11-310010

(22)出願日 平成11年10月29日(1999.10.29)

(71)出願人 000005049

シャープ株式会社

大阪府大阪市阿倍野区長池町22番22号

(72)発明者 田中 義朗

大阪府大阪市阿倍野区長池町22番22号 シ  
ャープ株式会社内

(74)代理人 100075557

弁理士 西教 圭一郎

Fターム(参考) 5B014 FA17 HC06

5B076 AC07 BB06

5B089 GA01 GA11 HA10 JB07 JB22

KA04 KA08 KA13 KB04 KB11

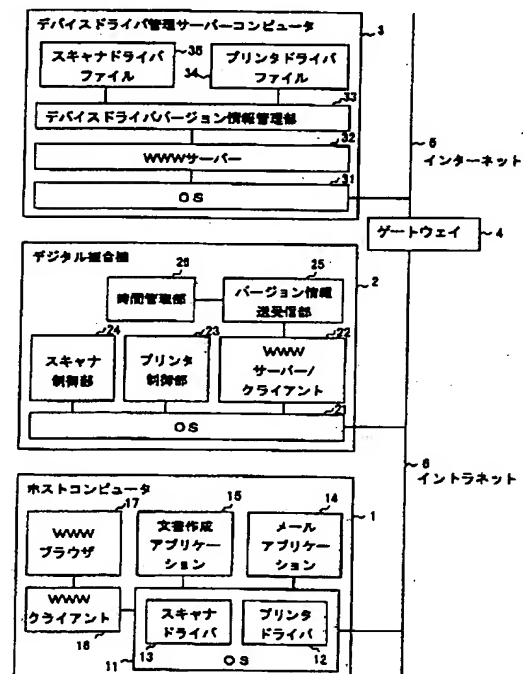
KC30 MB01 MC07

(54)【発明の名称】 ネットワーク接続された情報処理装置の周辺機器およびデバイスドライバのバージョン管理方法

## (57)【要約】

【課題】 ネットワーク接続された複数の情報処理装置が周辺機器を使用するためのデバイスドライバに対するバージョンアップを、効率的かつ確実に行うことが可能にする。

【解決手段】 イン트라ネット6を介して複数のホストコンピュータ1が接続され、周辺機器としてのデジタル複合機2を共用することができる。各ホストコンピュータ1は、OS11のデバイスドライバとして、デジタル複合機2を利用するためのプリンタドライバ12およびスキャナドライバ13を備える。これらのデバイスドライバのバージョン情報は、デジタル複合機2のバージョン情報送受信部25が管理し、時間管理部26の制御に基づいて、一定のタイミングでデバイスドライバ管理サーバーコンピュータ3にバージョン情報の問合せを行う。バージョン情報の問合せ結果は、各ホストコンピュータ1にイントラネット6を介して通知される。



## 【特許請求の範囲】

【請求項1】 内部のネットワークを介して複数の情報処理装置が接続され、それらの情報処理装置にインストールされているデバイスドライバによって制御される周辺機器において、

外部へのネットワークを介して、該デバイスドライバの管理を行う管理サーバー装置に接続され、該管理サーバー装置に保持されているデバイスドライバのバージョン情報の問合せを行い、バージョン情報を取得する問合せ手段と、

問合せ手段によって取得されるバージョン情報を、前記情報処理装置に通知する通知手段とを含むことを特徴とするネットワーク接続された情報処理装置の周辺機器。

【請求項2】 前記問合せ手段は、前記サーバー装置に対して、デバイスドライバのバージョン情報を、一定時間で問合せさせるタイマ手段を有することを特徴とする請求項1記載のネットワーク接続された情報処理装置の周辺機器。

【請求項3】 前記問合せ手段は、前記管理サーバー装置に対して、次回にデバイスドライバのバージョン情報を問合せすべき間隔を獲得する間隔獲得手段と、

該問合せ手段が問合せを行う間隔を、間隔獲得手段によって獲得された間隔に変更させる間隔変更手段とを有することを特徴とする請求項1記載のネットワーク接続された情報処理装置の周辺機器。

【請求項4】 前記問合せ手段は、前記サーバー装置に対して、電源立ち上げを検出して、立ち上げ時のみデバイスドライバのバージョン情報を問合せさせる立ち上げ検出手段を有することを特徴とする請求項1記載のネットワーク接続された情報処理装置の周辺機器。

【請求項5】 前記情報処理装置が使用しているデバイスドライバのバージョン情報を格納するバージョン格納手段と、

バージョン格納手段に格納されているバージョン情報と前記問合せ手段による管理サーバー装置への問合せで得られたデバイスドライバのバージョン情報との相違を判定するバージョン判定手段と、

バージョン判定手段による判定結果に相違がある場合のみ、該問合せで得られたバージョン情報を、前記通知手段によって情報処理装置に通知させる通知制御手段とを有することを特徴とする請求項1～4のいずれかに記載のネットワーク接続された情報処理装置の周辺機器。

【請求項6】 前記情報処理装置から、前記バージョン情報の通知要求を受信し、前記通知手段に、要求のある情報処理装置にのみバージョン情報を通知させるバージョン管理手段を有することを特徴とする請求項1～5のいずれかに記載のネットワーク接続された情報処理装置の周辺機器。

【請求項7】 前記デバイスドライバを保存することができる保存手段と、

前記情報処理装置が使用しているデバイスドライバのバージョン情報を格納するバージョン格納手段と、

前記バージョン格納手段に格納されているバージョン情報と前記問合せ手段による管理サーバー装置への問合せで得られたデバイスドライバのバージョン情報との相違を判定するバージョン判定手段と、

バージョン判定手段による判定結果に相違がある場合

に、該デバイスドライバを該管理サーバー装置から保存手段にダウンロードするダウンロード手段とを有することを特徴とする請求項1～6のいずれかに記載のネットワーク接続された情報処理装置の周辺機器。

【請求項8】 前記通知手段は、前記バージョン判定手段による判定結果で、前記バージョン格納手段に格納されている情報処理装置が使用しているデバイスドライバのバージョン情報と、前記問合せ手段によって管理サーバー装置から得られたデバイスドライバのバージョン情報との相違が生じた場合に、該情報処理装置に前記ダウンロード手段によってダウンロードされたデバイスドライバのインストールを実行するか否かの問合せを行うことを特徴とする請求項7記載のネットワーク接続された情報処理装置の周辺機器。

【請求項9】 前記デバイスドライバが複数のファイルによって構成され、デバイスドライバを構成するファイルがストアされる複数の管理サーバー装置のアドレスを、ファイルに対応して保持するアドレス保持手段をさらに含み、

前記ダウンロード手段は、デバイスドライバを構成する各ファイルを、アドレス保持手段によって保持される各ファイルに対応するアドレスからダウンロードすることを特徴とする請求項7または8記載のネットワーク接続された情報処理装置の周辺機器。

【請求項10】 前記問合せ手段は、予め設定されている管理サーバー装置のアドレスのログインパスワードを使って、該管理サーバー装置にアクセスすることを特徴とする請求項1～9のいずれかに記載のネットワーク接続された情報処理装置の周辺機器。

【請求項11】 前記問合せ手段は、

次回アクセスする場合のログインパスワードを前記管理サーバー装置から獲得するパスワード獲得手段と、パスワード獲得手段によって獲得されたログインパスワードを保持するパスワード保持手段とを有することを特徴とする請求項10記載のネットワーク接続された情報処理装置の周辺機器。

【請求項12】 前記問合せ手段は、

前記管理サーバー装置から、OS環境調査プログラムを獲得する調査プログラム獲得手段と、前記情報処理装置からの要求にตอบสนองして、調査プログラム獲得手段が獲得したOS環境調査プログラムを該情報

処理装置へ送付する調査プログラム送付手段と、  
該情報処理装置から該OS環境調査プログラムによる調査結果を獲得する調査結果獲得手段と、  
調査結果獲得手段によって獲得された調査結果を該管理サーバー装置に送付する調査結果送付手段とを含み、  
前記ダウンロード手段は、調査結果送付手段によって送付された調査結果に対応して該管理サーバー装置から提供されるデバイスドライバをダウンロードすることを特徴とする請求項7または8記載のネットワーク接続された情報処理装置の周辺機器。

【請求項13】 前記問合せ手段は、  
前記管理サーバー装置に保存されている複数のバージョンのデバイスドライバのバージョン情報を取得するバージョン情報獲得手段と、  
前記情報処理装置にインストールさせるデバイスドライバのバージョンを選択するバージョン選択手段とを含み、  
前記ダウンロード手段は、バージョン選択手段によって選択されたバージョンのデバイスドライバを管理サーバー装置からダウンロードすることを特徴とする請求項7または8記載のネットワーク接続された情報処理装置の周辺機器。

【請求項14】 前記保存手段は、複数のOS用のデバイスドライバを保存することを特徴とする請求項12または13記載のネットワーク接続された情報処理装置の周辺機器。

【請求項15】 前記ダウンロード手段は、前記管理サーバー装置から暗号処理を施されたデバイスドライバをダウンロードし、該暗号を解読する暗号解読手段を有することを特徴とする請求項7～14のいずれかに記載のネットワーク接続された情報処理装置の周辺機器。

【請求項16】 前記問合せ手段は、  
当該周辺機器の構成情報を、前記管理サーバー装置に対して送信する構成情報送信手段と、  
構成情報送信手段が送信した当該機器の構成情報に対応したファイルリストを、該管理サーバー装置から受信するファイルリスト受信手段とを有することを特徴とする請求項7～15のいずれかに記載のネットワーク接続された情報処理装置の周辺機器。

【請求項17】 前記問合せ手段は、  
前記管理サーバー装置への問合せ時に、前記ホスト装置にインストールされたデバイスドライバのOS情報と該デバイスドライバのバージョン情報とを、該管理サーバー装置に通知することを特徴とする請求項1～16のいずれかに記載のネットワーク接続された情報処理装置の周辺機器。

【請求項18】 前記情報処理装置から所定時間アクセスがないことを検知するアクセス無し検知手段を含み、  
前記ダウンロード手段は、アクセス無し検知手段の検知結果に応答して、前記デバイスドライバのダウンロード

を開始することを特徴とする請求項7～17のいずれかに記載のネットワーク接続された情報処理装置の周辺機器。

【請求項19】 前記問合せ手段は、前記サーバー装置に対して、HTTPを用いてデバイスドライバのバージョン情報を問合せることを特徴とする請求項1～18のいずれかに記載のネットワーク接続された情報処理装置の周辺機器。

【請求項20】 前記通知手段は、HTTPを用いて前記情報処理装置への要求および通知を行うことを特徴とする請求項1～19のいずれかに記載のネットワーク接続された情報処理装置の周辺機器。

【請求項21】 内部のネットワークを介して複数の情報処理装置が周辺機器に接続され、該周辺機器を制御するために各情報処理装置にインストールされているデバイスドライバのバージョンを管理するための方法において、

該周辺機器を、外部へのネットワークを介して該デバイスドライバの管理を行う管理サーバー装置に接続して、  
該周辺機器から、該管理サーバー装置に保持されているデバイスドライバのバージョン情報の問合せを行い、  
該管理サーバー装置から得られたバージョン情報を、該周辺機器から各該情報処理装置に通知することを特徴とするデバイスドライバのバージョン管理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、パーソナルコンピュータやワークステーションなどの情報処理装置とネットワークを介して接続されるプリンタなどの周辺機器と、当該周辺機器を情報処理装置で使用するためのデバイスドライバの管理方法に関する。

【0002】

【従来の技術】従来から、パーソナルコンピュータやワークステーションなどの情報処理装置は、種々の周辺機器を接続して、情報処理装置としての適用範囲を拡張することが可能となるように構成されている。周辺機器としては、情報処理装置に対して各種情報を入力したり、情報処理装置から得られる情報を出力したりする入出力装置などが代表的に用いられる。情報処理装置では、人間が情報処理装置を使用するための基本的な入出力は、オペレーションシステム（以下、「OS」と略称する）が受け持ち、OS上で種々のアプリケーションソフトウェアを実行するようにしている場合が多い。個々の情報処理装置毎に異なるハードウェアに関する処理は、OSが行い、アプリケーションソフトウェアは、OSが提供する機能を利用して、個々の情報処理装置のハードウェアの違いの影響を受けにくくすることができるからである。

【0003】しかしながら、情報処理装置に接続される周辺機器は、多種多様であり、それらの周辺機器を効率

的に動作させるためには、それぞれの周辺機器に適合するように作成されている駆動プログラムであるデバイスドライバを用いる必要がある。情報処理装置に接続する可能性がある周辺機器のデバイスドライバをすべてOSが備えることは、膨大な記憶容量を要するので、実用的ではない。一般には、周辺機器を接続する際に、必要なデバイスドライバを組み込むインストールを行う方法が採用されている。

【0004】特開平8-161250には、周辺機器自体が、その周辺機器を利用するためのデバイスドライバを格納しておき、その周辺機器に接続される情報処理装置であるホストコンピュータに、格納しているデバイスドライバをインストールする構成が提案されている。また、特開平10-301884では、周辺機器内にデバイスドライバとともに、ホストコンピュータのOS環境を調査するプログラムを備えておき、ホストコンピュータのハードウェアやソフトウェアの環境に合ったデバイスドライバを、周辺機器からホストコンピュータに送信する構成が開示されている。

【0005】

【課題を解決しようとする課題】コンピュータが使用するプログラムは、一旦作成されたあとでも、不具合が発見されてその是正が行われたり、性能を向上したりするために、改良が施されることが多い。一般的にソフトウェアではバージョン管理がなされ、変更がバージョンに反映されるので、できるだけ最新のバージョンのソフトウェアを使用することが推奨されている。

【0006】周辺機器のデバイスドライバも、一般的に広く使用されているOSを対象として、周辺機器のメーカーなどから提供され、改良に伴うバージョン管理もなされている。デバイスドライバについてプログラムの改良などが行われると、フロッピーディスクやCD-ROMなどの記録媒体に記録された状態でユーザに送付されたり、インターネットなどのネットワークを介してユーザに提供したりすることが行われている。

【0007】近年、1つの周辺機器を、複数の情報処理装置で共同で使用するために、ネットワークを介して接続する構成が利用されている。このようなネットワーク構成では、周辺機器を利用するためのデバイスドライバを、各情報処理装置のOSにインストールする必要がある。同一の周辺機器に対するデバイスドライバに改良が施されると、それまでのデバイスドライバと異なるバージョンのデバイスドライバを各情報処理装置にインストールされる必要がある。従来一般的には、ネットワークシステムの管理者が、各情報処理装置に、改良されたバージョンのデバイスドライバをインストールし直しているけれども、その管理の手間は多大なものとなる。特開平8-161250や特開平10-301884などの先行技術では、周辺機器から自動的にホストコンピュータにデバイスドライバを供給してインストールすること

ができる。

【0008】しかしながら、デバイスドライバやOSなどのソフトウェアは、バージョンアップや新しいコンセプトに基づいた新商品が発表されるため、予め周辺機器に用意されているデバイスドライバだけでは対応することができなくなる事態が生じてしまう。

【0009】本発明の目的は、デバイスドライバのバージョンアップやホストコンピュータとなる情報処理装置のハードウェアやソフトウェアの環境の進化に対応して、情報処理装置にインストールされているデバイスドライバの管理を効率的に行うことができるネットワーク接続された情報処理装置の周辺機器およびデバイスドライバの管理方法を提供することである。

【0010】

【課題を解決するための手段】本発明は、内部のネットワークを介して複数の情報処理装置が接続され、それらの情報処理装置にインストールされているデバイスドライバによって制御される周辺機器において、外部へのネットワークを介して、該デバイスドライバの管理を行う管理サーバー装置に接続され、該管理サーバー装置に保持されているデバイスドライバのバージョン情報の問合せを行い、バージョン情報を取得する問合せ手段と、問合せ手段によって取得されるバージョン情報を、前記情報処理装置に通知する通知手段とを含むことを特徴とするネットワーク接続された情報処理装置の周辺機器である。

【0011】本発明に従えば、複数の情報処理装置は内部のネットワークを介して周辺機器に接続される。各情報処理装置は、インストールされているデバイスドライバによって周辺機器を制御する。周辺機器は、問合せ手段と通知手段とを含む。問合せ手段は、外部へのネットワークを介して、デバイスドライバの管理を行う管理サーバー装置に接続され、管理サーバー装置に保持されているデバイスドライバのバージョン情報を問合せ、バージョン情報を取得する。通知手段は、問合せ手段によって取得されるバージョン情報を、情報処理装置に通知する。各情報処理装置を利用するユーザは、当該情報処理装置が使用しているデバイスドライバが最新のバージョンであるか否かを判定することができる。

【0012】また本発明で、前記問合せ手段は、前記サーバー装置に対して、デバイスドライバのバージョン情報を、一定時間で問合せさせるタイマ手段を有することを特徴とする。

【0013】本発明に従えば、タイマ手段によって、問合せ手段による管理サーバー装置に対してのデバイスドライバのバージョン情報の問合せを、一定時間間隔で行わせるので、一定時間間隔、たとえば毎日の定時や週に1度などで問合せを行い、情報処理装置のユーザはインストールされているデバイスドライバが最新のものであるか否かを絶えず監視することができる。



【0014】また本発明で、前記問合せ手段は、前記管理サーバー装置に対して、次回にデバイスドライバのバージョン情報を問合せすべき間隔を獲得する間隔獲得手段と、該問合せ手段が問合せを行う間隔を、間隔獲得手段によって獲得された間隔に変更させる間隔変更手段とを有することを特徴とする。

【0015】本発明に従えば、問合せ手段による管理サーバー装置へのバージョン情報の問合せは、間隔獲得手段が管理サーバー装置から獲得する間隔に応じて、間隔変更手段が変更するので、管理サーバー装置側でデバイスドライバのバージョンアップのスケジュールが予め判明しているときには、スケジュールに合わせて間隔を調整させ、バージョンアップのスケジュールに合せた問合せが適切に行われるようにすることができる。バージョンアップが行われていないタイミングで周辺機器から外部のネットワークを介して管理サーバー装置への問合せがなくなるので、管理サーバー装置へのトラフィックを軽減することができる。

【0016】また本発明で、前記問合せ手段は、前記管理サーバー装置に対して、電源立ち上げを検出して、立ち上げ時のみデバイスドライバのバージョン情報を問合せさせる立ち上げ検出手段を有することを特徴とする。

【0017】本発明に従えば、立ち上げ検出手段が電源立ち上げを検出しているときに、問合せ手段は管理サーバー装置に対してバージョン情報の問合せを行うので、周辺機器は、管理サーバー装置に対して、電源立ち上げに伴うウォーミングアップ中にバージョン情報を問合せすることができる。ネットワークに接続される複数の情報処理装置からの周辺機器の利用は、ウォーミングアップ中には行われる可能性が少ないので、バージョン情報の問合せを、ネットワーク全体としての効率などに影響を与えないタイミングで行うことができる。電源の立ち上げの検出は比較的簡単な構成で行うことができるので、一定の時間間隔で問合せを行うような管理よりも、低コストでバージョン情報の管理を行うことができる。

【0018】また本発明は、前記情報処理装置が使用しているデバイスドライバのバージョン情報を格納するバージョン格納手段と、バージョン格納手段に格納されているバージョン情報と前記問合せ手段による管理サーバー装置への問合せで得られたデバイスドライバのバージョン情報との相違を判定するバージョン判定手段と、バージョン判定手段による判定結果に相違がある場合のみ、該問合せで得られたバージョン情報を、前記通知手段によって情報処理装置に通知させる通知制御手段とを有することを特徴とする。

【0019】本発明に従えば、周辺機器には、ネットワークを介して接続される複数の情報処理装置が使用しているデバイスドライバのバージョン情報を格納するバージョン格納手段と、格納されているバージョン情報と、管理サーバー装置への問合せで得られるデバイスドライ

バのバージョン情報との相違を判定するバージョン判定手段と、判定結果に相違がある場合のみ問合せで得られたバージョン情報を通知手段によって情報処理装置に通知させる通知制御手段とを含む。情報処理装置が使用しているデバイスドライバのバージョン情報を予め格納して管理しておき、管理サーバー装置に保存されているデバイスドライバのバージョン情報との相違を判定し、相違がある場合のみ情報処理装置に通知することによって、内部のネットワークを介する通知のトラフィックを減らすことができる。

【0020】また本発明は、前記情報処理装置から、前記バージョン情報の通知要求を受信し、前記通知手段に、要求のある情報処理装置にのみバージョン情報を通知させるバージョン管理手段を有することを特徴とする。

【0021】本発明に従えば、周辺機器は、ネットワークを介して接続される情報処理装置から、バージョン情報の通知要求を受信し、通知手段に対して、要求がある情報処理装置にのみバージョン情報を通知させるバージョン管理手段を有する。通知要求がある情報処理装置にのみバージョン情報を通知するので、不特定多数の情報処理装置にバージョン情報を通知するのと比較して、内部のネットワーク環境でのトラフィックを減らすことができる。バージョンアップの必要性は、たとえばアプリケーションソフトウェアを新たに導入する情報処理装置で、そのアプリケーションソフトウェアによってバージョン情報の通知が要求される場合があり、そのようなアプリケーションソフトウェアを導入しない情報処理装置では、特にバージョン情報の通知は不要であることも多い。バージョンアップの必要のない情報処理装置に対してバージョン情報を通知しないので、トラフィック量の軽減を図ることができる。

【0022】また本発明は、前記デバイスドライバを保存することができる保存手段と、前記情報処理装置が使用しているデバイスドライバのバージョン情報を格納するバージョン格納手段と、前記バージョン格納手段に格納されているバージョン情報と前記問合せ手段による管理サーバー装置への問合せで得られたデバイスドライバのバージョン情報との相違を判定するバージョン判定手段と、バージョン判定手段による判定結果に相違がある場合に、該デバイスドライバを該管理サーバー装置から保存手段にダウンロードするダウンロード手段とを有することを特徴とする。

【0023】本発明に従えば、周辺機器は、デバイスドライバを保存することができる保存手段と、情報処理装置が使用しているデバイスドライバのバージョン情報を格納するバージョン格納手段と、格納されているバージョン情報と管理サーバー装置への問合せで得られるバージョン情報との相違を判定するバージョン判定手段と、バージョンの判定結果に相違がある場合にデバイスドライ

イバを管理サーバー装置からダウンロードするダウンロード手段とを有するので、バージョンアップが行われているときにのみ最新のデバイスドライバを周辺機器にダウンロードして保存手段に保存することができる。各情報処理装置では、周辺機器の保存手段から、最新のデバイスドライバを内部のネットワークを介してダウンロードすることができ、各情報処理装置でのデバイスドライバのバージョンアップを容易に行うことができる。

【0024】また本発明で、前記通知手段は、前記バージョン判定手段による判定結果で、前記バージョン格納手段に格納されている情報処理装置が使用しているデバイスドライバのバージョン情報と、前記問合せ手段によって管理サーバー装置から得られたデバイスドライバのバージョン情報との相違が生じた場合に、該情報処理装置に前記ダウンロード手段によってダウンロードされたデバイスドライバのインストールを実行するか否かの問合せを行うことを特徴とする。

【0025】本発明に従えば、情報処理装置が使用しているデバイスドライバのバージョン情報と、管理サーバー装置からの問合せで得られるバージョン情報とに差異が生じると、通知手段はダウンロード手段によってダウンロードされたデバイスドライバのインストールを実行するか否かを情報処理装置に問合せする。各情報処理装置のユーザは、デバイスドライバのバージョンアップが行われて、最新のデバイスドライバが周辺機器に保存されていることを知ることができるので、必要に応じてインストールするか否かの判断を行うことができる。

【0026】また本発明で、前記デバイスドライバが、複数のファイルによって構成され、デバイスドライバを構成するファイルがストアされる複数の管理サーバー装置のアドレスを、ファイルに対応して保持するアドレス保持手段をさらに含み、前記ダウンロード手段は、デバイスドライバを構成する各ファイルを、アドレス保持手段によって保持される各ファイルに対応するアドレスからダウンロードすることを特徴とする。

【0027】本発明に従えば、デバイスドライバを複数のファイルによって構成しておき、複数の管理サーバー装置でファイルを分散して管理するので、管理サーバー装置へのトラフィックを軽減することが可能となり、バージョンアップに伴う最新のデバイスドライバの配布を高速に行うことができる。

【0028】また本発明で、前記問合せ手段は、予め設定されている管理サーバー装置のアドレスのログインパスワードを使って、該管理サーバー装置にアクセスすることを特徴とする。

【0029】本発明に従えば、周辺機器が管理サーバー装置に問合せを行う際には、管理サーバー装置のアドレスのログインパスワードを使ってアクセスするので、ログインパスワードを取得していない周辺機器や情報処理装置からのアクセスを防いでセキュリティを向上させる

ことができる。

【0030】また本発明で、前記問合せ手段は、次回アクセスする場合のログインパスワードを前記管理サーバー装置から獲得するパスワード獲得手段と、パスワード獲得手段によって獲得されたログインパスワードを保持するパスワード保持手段とを有することを特徴とする。

【0031】本発明に従えば、周辺機器の問合せ手段が管理サーバー装置にアクセスしたあとで次のアクセスを行う場合のログインパスワードは、管理サーバー装置からパスワード獲得手段によって獲得され、パスワード保持手段によって保持されるものを使用する。これによって、管理サーバー装置は各周辺機器毎に個別にパスワードを設けることが可能となり、不正なアクセスを有効に防止することができる。

【0032】また本発明で、前記問合せ手段は、前記管理サーバー装置から、OS環境調査プログラムを獲得する調査プログラム獲得手段と、前記情報処理装置からの要求にตอบสนองして、調査プログラム獲得手段が獲得したOS環境調査プログラムを該情報処理装置へ送付する調査プログラム送付手段と、該情報処理装置から該OS環境調査プログラムによる調査結果を獲得する調査結果獲得手段と、調査結果獲得手段によって獲得された調査結果を該管理サーバー装置に送付する調査結果送付手段とを含み、前記ダウンロード手段は、調査結果送付手段によって送付された調査結果に対応して該管理サーバー装置から提供されるデバイスドライバをダウンロードすることを特徴とする。

【0033】本発明に従えば、周辺機器の問合せ手段は、調査プログラム獲得手段によって管理サーバー装置からOS環境調査プログラムを獲得し、調査プログラム送付手段によって、情報処理装置からの要求に応じて獲得した調査プログラムを情報処理装置に送付する。調査結果獲得手段は、情報処理装置から、OS環境調査プログラムによる調査結果を獲得する。調査結果送付手段は、獲得した調査結果を管理サーバー装置に送付する。ダウンロード手段は、調査結果に対応して管理サーバー装置から提供されるデバイスドライバをダウンロードする。これによって、ネットワークを介して接続される情報処理装置のハードウェアやソフトウェアに適合したデバイスドライバのバージョンアップが可能となる。また、OS環境調査プログラムは、管理サーバー装置から提供されるので、情報処理装置が搭載するOSなどが新しくなっても、管理サーバー装置にそのOS用のOS環境調査プログラムを用意すればよく、周辺機器に予め格納しておくだけでは対応することができない場合でも、有効に利用することができるOS環境調査プログラムを取得することができる。

【0034】また本発明で、前記問合せ手段は、前記管理サーバー装置に保存されている複数のバージョンのデバイスドライバのバージョン情報を取得するバージョン

情報獲得手段と、前記情報処理装置にインストールさせるデバイスドライバのバージョンを選択するバージョン選択手段とを含み、前記ダウンロード手段は、バージョン選択手段によって選択されたバージョンのデバイスドライバを管理サーバー装置からダウンロードすることを特徴とする。

【0035】本発明に従えば、周辺機器の問合せ手段は、管理サーバー装置に保存されている複数のバージョンのデバイスドライバについてのバージョン情報をバージョン情報獲得手段によって取得し、バージョン選択手段によって、ネットワークを介して接続される情報処理装置にインストールさせるデバイスドライバのバージョンを選択する。ダウンロード手段は、バージョン選択手段によって選択されたバージョンのデバイスドライバを管理サーバー装置からダウンロードするので、情報処理装置に対して適合したバージョンのデバイスドライバをダウンロードすることができる。一般にデバイスドライバのバージョンアップでは、機能が追加され、その結果プログラムサイズも大きくなり、記憶容量の必要性も増大する。情報処理装置の環境に応じてインストールさせるバージョンを選択することによって、広範囲な情報処理装置への対応が可能となる。

【0036】また本発明で、前記保存手段は、複数のOS用のデバイスドライバを保存することを特徴とする。

【0037】本発明に従えば、周辺機器の保存手段には、複数のOS用のデバイスドライバを保存するので、周辺機器に対してネットワークを介して接続される情報処理装置で、複数種類のOSが使用されても、周辺機器からデバイスドライバを提供して、管理サーバー装置へのトラフィックを軽減することができる。

【0038】また本発明で、前記ダウンロード手段は、前記管理サーバー装置から暗号処理を施されたデバイスドライバをダウンロードし、該暗号を解読する暗号解読手段を有することを特徴とする。

【0039】本発明に従えば、管理サーバー装置から周辺機器にデバイスドライバをダウンロードするダウンロード手段は、管理サーバー装置で暗号処理を施されたデバイスドライバをダウンロードし、暗号解読手段によって暗号を解読するので、ダウンロード中にデバイスドライバの内容が変更されていないことが保証される。暗号の解読は周辺機器内のダウンロード手段が行うので、ネットワークを介して接続される情報処理装置では暗号の解読を行う必要はなく、保存手段に保存されているデバイスドライバをそのまま利用することができる。

【0040】また本発明で、前記問合せ手段は、当該周辺機器の構成情報を、前記管理サーバー装置に対して送信する構成情報送信手段と、構成情報送信手段が送信した当該機器の構成情報に対応したファイルリストを、該管理サーバー装置から受信するファイルリスト受信手段とを有することを特徴とする。

【0041】本発明に従えば、周辺機器の問合せ手段が管理サーバー装置に対して問合せを行う際には、周辺機器の構成情報を構成情報送信手段によって送信し、送信した当該周辺機器の構成情報に対応したファイルリストを管理サーバー装置からファイルリスト受信手段によって受信する。管理サーバー装置は、周辺機器の構成情報送信手段から送付される構成情報に対応してファイルリストを作成すればよいので、周辺機器の構成に応じてデバイスドライバを予め格納しておく必要はなく、管理サーバー装置での記憶容量を節約することができる。

【0042】また本発明で、前記問合せ手段は、前記管理サーバー装置への問合せ時に、前記ホスト装置にインストールされたデバイスドライバのOS情報と該デバイスドライバのバージョン情報とを、該管理サーバー装置に通知することを特徴とする。

【0043】本発明に従えば、周辺機器の問合せ手段による管理サーバー装置への問合せ時に、情報処理装置にインストールされているデバイスドライバのOS情報と、デバイスドライバのバージョン情報とを通知するので、情報処理装置で使用されている頻度が高いOSなどについての情報を得ることができ、デバイスドライバの開発を進める必要があるOSを割り出すことができる。

【0044】また本発明は、前記情報処理装置から所定時間アクセスがないことを検知するアクセス無し検知手段を含み、前記ダウンロード手段は、アクセス無し検知手段の検知結果に応答して、前記デバイスドライバのダウンロードを開始することを特徴とする。

【0045】本発明に従えば、周辺機器が管理サーバー装置からデバイスドライバのダウンロードを開始するのは、アクセス無し検知手段によって情報処理装置から周辺機器に対して所定時間アクセスがないことを検知した結果に基づいて行うので、内部のネットワーク環境での周辺機器に対するアクセスが少ない状態でデバイスドライバのダウンロードを行い、高速にデバイスドライバをダウンロードすることができる。

【0046】また本発明で、前記問合せ手段は、前記サーバー装置に対して、HTTPを用いてデバイスドライバのバージョン情報を問合せることを特徴とする。

【0047】本発明に従えば、周辺機器の問合せ手段は、管理サーバー装置に対して、HTTPを用いてデバイスドライバのバージョン情報を問合せるので、内部のネットワークの管理者のみならず、内部のネットワークと外部へのネットワークとの間に設けられているゲートウェイなどに導入されているセキュリティシステムを通過して、外部の管理サーバー装置にバージョン情報の問合せを行うことができる。

【0048】また本発明で、前記通知手段は、HTTPを用いて前記情報処理装置への要求および通知を行うことを特徴とする。

【0049】本発明に従えば、通知手段がネットワーク

を介して周辺機器に接続されている情報処理装置に対する要求および通知を行う際には、HTTPを用いるので、外部の管理サーバー装置に対して問合せを行う際に用いるプロトコルと同じプロトコルで内部のネットワークを介して接続される情報処理装置に対しても要求や通知を行うことができ、すべて同じプロトコルで処理することができるので、情報処理装置と管理サーバー装置とにシームレスに接続することが可能となる。

【0050】さらに本発明は、内部のネットワークを介して複数の情報処理装置が周辺機器に接続され、該周辺機器を制御するために各情報処理装置にインストールされているデバイスドライバのバージョンを管理するための方法において、該周辺機器を、外部へのネットワークを介して該デバイスドライバの管理を行う管理サーバー装置に接続して、該周辺機器から、該管理サーバー装置に保持されているデバイスドライバのバージョン情報の問合せを行い、該管理サーバー装置から得られたバージョン情報を、該周辺機器から各情報処理装置に通知することを特徴とするデバイスドライバのバージョン管理方法である。

【0051】本発明に従えば、周辺機器メーカーなどによって設置され、常に最新のバージョンのデバイスドライバが保存されている管理サーバー装置に対して、周辺機器はネットワークを介して接続される情報処理装置に代わってデバイスドライバのバージョン情報を問合せ、周辺機器に接続されている情報処理装置に通知することによって、情報処理装置のユーザは情報処理装置が使用しているデバイスドライバが最新のバージョンであるか否かを判定することができる。

#### 【0052】

【発明の実施の形態】以下、図面に基づいて、本発明の実施の各形態の構成や動作を説明する。以下の説明で、先行して説明した部分に対応する部分には同一の参照符を付し、重複する説明を省略する。

【0053】図1は、本発明の実施の第1形態としてのネットワークシステム構成を示す。本実施形態のネットワークシステムは、基本的に複数の情報処理装置であるホストコンピュータ1と、複数のホストコンピュータ1によって共用可能な周辺機器であるデジタル複合機2と、各ホストコンピュータ1でデジタル複合機2を使用するためのデバイスドライバを管理する管理サーバー装置であるデバイスドライバ管理サーバーコンピュータ3を含む。デジタル複合機2は、ゲートウェイ4を介して接続される外部へのネットワークとしてのインターネット5を経て、デバイスドライバ管理サーバーコンピュータ3にアクセスすることができる。デジタル複合機2と複数のホストコンピュータ1とは、内部のネットワークであるイントラネット6を介して接続される。イントラネット6は、たとえばLANなど、1つの事業体内部に設けられるネットワークであり、外部からの不正なア

クセスに対して防衛されている。インターネット5は、広く外部に対して開放されている。インターネット5とイントラネット6とは、同じHTTP (Hyper Text Transfer Protocol) と呼ばれる通信プロトコルを用いて通信を行う。したがって、イントラネット6内での通信も、ゲートウェイ4を介してのインターネット5に接続した通信も、同様に行うことができる。ただし、ゲートウェイ4には、イントラネット6でのセキュリティを確保するためのファイアウォールなどが設けられ、インターネット5を介して外部からイントラネット6に不正なアクセスが行われることがないように、セキュリティシステムが構築されている。

【0054】ホストコンピュータ1は、OS11を備え、デジタル複合機2を使用するためのプリンタドライバ12やスキャナドライバ13を、OS11の一部を構成するデバイスドライバとして組込んでいる。ホストコンピュータ1を利用するユーザは、メールアプリケーション14や文書作成アプリケーション15などを利用する際に、OS11に搭載されているプリンタドライバ12やスキャナドライバ13を間接的に利用する。ホストコンピュータ1には、インターネット5を利用して通信を行うためのWWW (World Wide Web) クライアント16も備えられ、閲覧用のWWWブラウザ17を用いて、Webページのグラフィック表示を行わせることができる。メールアプリケーション14や文書作成アプリケーション15などを利用して作成された文書や受信された文書は、イントラネット6を介してデジタル複合機2に送信し、デジタル複合機2の有するプリンタとしての機能で印刷することができる。

【0055】デジタル複合機2は、プリンタとしての機能とスキャナとしての機能とが、イントラネット6を介してホストコンピュータ1から利用可能である。デジタル複合機2は、単独で、コピー装置やファクシミリ装置としての機能を備えることもできる。デジタル複合機2は、OS21を備え、プリンタおよびスキャナとしての動作は、OS21によって制御されるプリンタ制御部23およびスキャナ制御部24によってそれぞれ行われる。デジタル複合機2が、イントラネット6やインターネット5を介してホストコンピュータ1やデバイスドライバ管理サーバーコンピュータ3と通信を行うために、WWWサーバー/クライアント22が設けられている。デジタル複合機2が、ホストコンピュータ1に搭載されているプリンタドライバ12やスキャナドライバ13などのデバイスドライバのバージョンを管理するための機能として、デジタル複合機2はバージョン情報送受信部25や、時間管理部26を備えている。バージョン情報送受信部25は、WWWサーバー/クライアント22を介して、ホストコンピュータ1やデバイスドライバ管理サーバーコンピュータ3と通信を行う。時間管理部26は、バージョン情報送受信部25が行う送受信タイミン

グなどを制御する。

【0056】 デバイスドライバ管理サーバーコンピュータ3は、OS31に基づいて各種制御を行い、インターネット5を介してデジタル複合機2やホストコンピュータ1と通信を行うためのWWWサーバー32を備えている。デジタル複合機2やホストコンピュータ1は、WWWサーバー32を介して、デバイスドライババージョン情報管理部33や、プリンタドライバファイル34およびスキャナドライバファイル35にアクセスすることができる。プリンタドライバファイル34およびスキャナドライバファイル35には、ホストコンピュータ1に搭載されるプリンタドライバ12およびスキャナドライバ13がそれぞれ格納される。プリンタドライバ12またはスキャナドライバ13に対してバージョンアップが行われると、バージョンアップされた新たなプリンタドライバ12やスキャナドライバ13がプリンタドライバファイル34またはスキャナドライバファイル35にそれぞれ格納される。デバイスドライババージョン情報管理部33には、プリンタドライバファイル34およびスキャナドライバファイル35に格納されるプリンタドライバ12やスキャナドライバ13のバージョン情報が格納される。プリンタドライバ12やスキャナドライバ13で、重大な欠陥が発見されたバージョンについては、プリンタドライバファイル34やスキャナドライバファイル35から削除される。

【0057】 図2は、図1の実施形態でのデバイスドライバのバージョン管理の手順を概略的に示す。デバイスドライバ管理サーバーコンピュータ3は、動作を開始後、ステップa10で待機しており、デジタル複合機2からの要求に応じて、ステップa11で、管理サーバー装置として、デバイスドライババージョン情報管理部33に格納されているデバイスドライバのバージョン情報を送信し、再びステップa10の待機に戻る。デジタル複合機2は、動作開始後、ステップa21で一定の準備等のための待機を行い、ステップa22で、問合せ手段としてのバージョン情報送受信部25がデバイスドライバ管理サーバーコンピュータ3のアドレスであるURL (Uniform Resource Locator) を指定してアクセスし、バージョン情報を要求する。ステップa23で、デバイスドライバのバージョン情報を受信すると、ステップa24で、通知手段としてのバージョン情報送受信部25が、デバイスドライバのバージョン情報を、各ホストコンピュータ1に送信し、ステップa21に戻る。各ホストコンピュータ1は、動作開始後、ステップa31で待機し、デジタル複合機2からデバイスドライバのバージョン情報が送信されると、ステップa32でデバイスドライバのバージョン情報を受信し、ステップa31に戻る。

【0058】 したがって、デジタル複合機2は、常にデバイスドライバ管理サーバーコンピュータ3にアクセス

してデバイスドライバのバージョン情報を要求しており、バージョン情報を受信すると、直ちに各ホストコンピュータ1に送信する。各ホストコンピュータ1では、常に最新バージョンのデバイスドライバの情報を得ることができ、ホストコンピュータ1を利用するユーザは使用しているデバイスドライバが最新のものであるかどうかを容易に判定することができる。

【0059】 また、デジタル複合機2は、デバイスドライバ管理サーバーコンピュータ3に対して、HTTPを用いてデバイスドライバのバージョン情報の問合せを行うので、イントラネット6のネットワーク管理者のみならず、イントラネット6と外部のインターネット5との間に設けられるゲートウェイ4に導入されているファイアウォールなどのセキュリティシステムを通過し、外部のデバイスドライバ管理サーバーコンピュータ3に対してバージョン情報の問合せを行うことができる。また、内部のイントラネット6を介するホストコンピュータ1への要求や通知も、HTTPを用いて行うので、すべて同じプロトコルで処理することができ、ホストコンピュータ1とデバイスドライバ管理サーバーコンピュータ3との間の接続を、シームレスで行うことができる。

【0060】 図3は、本発明の実施の第2形態としてのバージョン管理の手順を示す。本実施形態のシステム構成は、図1と同様である。図1のデジタル複合機2のバージョン情報送受信部25は、問合せ手段としてのバージョン情報受信部および通知手段としてのバージョン情報送信部として独立して動作する。バージョン情報受信部は、動作開始後、ステップb11で、一定時間の待機を行う。この一定時間は、タイマ手段として動作する時間管理部26によって制御される。一定時間は、たとえば毎日の一定時刻などのタイミングや、週に一度などの頻度として、予め設定される。ステップb11の待機が終了すると、ステップb12およびステップb13で、図2のステップa22およびステップa23の各ステップと同様に、デバイスドライバ管理サーバーコンピュータ3にバージョン情報の要求を行い、バージョン情報の受信を行う。ステップb14では、受信されたデバイスドライバのバージョン情報を、バージョン情報送受信部25内に設けられるメモリに格納する。

【0061】 バージョン情報送信部としての動作は、動作開始後、ステップb21で待機する。ステップb22では、バージョン情報送受信部25の内部のメモリに、デバイスドライバのバージョン情報が格納済みであるか否かを判断する。バージョン情報が格納済みでないと判断されるときには、ステップb21に戻る。ステップb22で、デバイスドライバのバージョン情報が格納済みであると判断されるときには、ステップb23で、デバイスドライバのバージョン情報をホストコンピュータ1に送信して、ステップb21に戻る。

【0062】 本実施形態では、ステップb12でのデバ

イスドライバ管理サーバーコンピュータ3へのアクセスが、時間管理部26によって設定される一定時間毎に行われるので、アクセスが頻繁に発生することがなく、インターネット5を介してのトラフィックを軽減することができる。

【0063】図4は、本発明の実施の第3形態としてのバージョン管理の手順を示す。本実施形態も、図1のシステム構成で動作する。バージョン情報受信部25は、図3の実施形態と同様に、バージョン情報受信部とバージョン情報送信部とに独立して動作する。バージョン情報受信部には、次回問合せ時間が設定される。バージョン情報受信部が動作を開始すると、ステップc11で、設定されている次回問合せ時間だけ待機する。待機後、ステップc12で、デバイスドライバ管理サーバーコンピュータ3のURLを指定してアクセスを行い、問合せ手段および間隔獲得手段として、バージョン情報および更新予定時間をそれぞれ要求する。ステップc13では、デバイスドライバ管理サーバーコンピュータ3から、デバイスドライバのバージョン情報およびバージョン情報更新予定時間を受信する。ステップc14では、デバイスドライバのバージョン情報を、バージョン情報受信部25に備えるメモリに格納する。ステップc15では、間隔変更手段として、バージョン情報更新予定時間に基づき、バージョン管理部26の次回問合せ時間を設定し、ステップc11に戻る。

【0064】バージョン情報送信部の動作は、動作開始後、ステップc21で待機し、ステップc22では、デバイスドライバのバージョン情報がバージョン情報受信部25内のメモリに格納済みであるか否かを判断する。格納済みでないと判断されるときには、ステップc21に戻る。ステップc22で、デバイスドライバのバージョン情報が格納済みであると判断されるときには、通知手段として、ステップc23で、デバイスドライバのバージョン情報をホストコンピュータ1に送信し、ステップc21に戻る。本実施形態の動作は、図3に示す実施形態に類似しているけれども、デバイスドライバのバージョンアップのスケジュールが予め解っている場合に、デジタル複合機2はデバイスドライバ管理サーバーコンピュータ3に、バージョン情報とともに、その次の更新予定時間を要求する点で相違している。デバイスドライバ管理サーバーコンピュータ3から、次のバージョン情報更新予定時間が受信されると、バージョン情報受信部の次回問合せ時間の設定が変更される。したがって、デバイスドライバ管理サーバーコンピュータ3へのアクセスは、必要最小限にしか発生せず、インターネット5を介する通信によるトラフィックを、一層軽減することができる。

【0065】図5は、本発明の実施の第4形態としてのバージョン管理の動作手順を示す。本実施形態も図1のシステム構成で動作する。本実施形態は、図2の実施形

態に類似するけれども、デジタル複合機2がデバイスドライバ管理サーバーコンピュータ3にアクセスするタイミングが、電源オンの検出時点である点が相違する。バージョン情報受信部25内には、電源の立ち上りを検知する立ち上り検知手段が設けられ、電源オンを検知すると、ステップd11で、デバイスドライバ管理サーバーコンピュータのURLを指定して、バージョン情報を要求する。デバイスドライバ管理サーバーコンピュータ3からデバイスドライバのバージョン情報が送信されると、ステップd12で受信する。ステップd13では、デバイスドライバのバージョン情報をホストコンピュータ1に送信する。以上のようなステップc11からステップc13までの各ステップは、図2のステップa22からステップa24と基本的に同等である。

【0066】本実施形態では、電源オンの時点で外部のデバイスドライバ管理サーバーコンピュータ3に対するアクセスを行う。デジタル複合機2が外部のインターネット5を介してデバイスドライバ管理サーバーコンピュータ3にアクセスする際には、内部のイントラネット6も使用するので、イントラネット6を介して接続されるホストコンピュータ1からのデジタル複合機2への利用は制限を受ける。電源の立ち上り時点では、デジタル複合機2がホストコンピュータ1から利用される可能性は少なく、ネットワークシステムとしての効率を低下させずに、バージョン情報の取得を行うことができる。電源オンの検出は、比較的簡単な構成で行うことができ、バージョン情報受信部25に備えることによって、図1のバージョン管理部26での時間管理などを行う必要をなくし、時間管理部26を省略すればコストダウンを図ることができる。

【0067】図6は、本発明の実施の第5形態としてのデジタル複合機2によるバージョン管理の動作手順を示す。本実施形態は、図2の実施形態のステップa21、ステップa22およびステップa23とそれぞれ同様なステップe11、ステップe12およびステップe13を経て、デバイスドライバ管理サーバーコンピュータ3からデバイスドライバのバージョンを取得する。本実施形態で、図2の実施形態と相違する点は、たとえばバージョン情報受信部25に現在ホストコンピュータ1で使用しているデバイスドライバのバージョン情報を格納するバージョン情報格納手段が設けられている点である。また、ステップe14では、バージョン判定手段として受信したデバイスドライバのバージョン情報が、現在保持しているバージョンと同じであるか否かを判定する。バージョンが同じであれば、ステップe11に戻る。ステップe14で、バージョン情報が同じでないと判定されるときには、通知制御手段として、ステップe15で、デバイスドライバ管理サーバーコンピュータ3から受信されたデバイスドライバのバージョン情報をホストコンピュータ1に送信し、ステップe16で、デバ



イスドライバのバージョン情報を更新するように制御し、ステップe11に戻る。

【0068】本実施形態では、デバイスドライバ管理サーバーコンピュータ3から得られたバージョン情報が、バージョン情報送受信部25に格納されているバージョン情報と一致している場合、すなわちデバイスドライバのバージョンが更新されていない場合には、ステップe15でのホストコンピュータ1への送信を行わず、バージョン情報が一致していない場合、すなわちデバイスドライバが更新されている場合のみステップe15で送信を行うとともに、更新されたバージョン情報を新たにステップe16で格納するようにしている。したがって、内部のイントラネット6などの、ユーザネットワーク環境でのトラフィックを減らすことができる。

【0069】図7は、本発明の実施の第6形態としての図1のシステムの動作手順を示す。本実施形態は図2に示す実施形態に類似し、ホストコンピュータ1からデジタル複合機2にバージョン情報の要求が行われるときのみ、デジタル複合機2はデバイスドライバ管理サーバーコンピュータ3に対してバージョン情報を要求し、受信されたバージョン情報を、バージョン情報の要求が行われたホストコンピュータ1にのみ送信することである。

【0070】すなわち、ステップf11からステップf12までのデバイスドライバ管理サーバーコンピュータ3の動作は、図2のステップa10およびステップa11の各ステップと同等である。デジタル複合機2の動作は、動作開始後、ステップf21で、待機する。待機終了後の、ステップf22、ステップf23およびステップf24の各ステップは、図2のステップa22、ステップa23およびステップa24の各ステップとそれぞれ同等である。

【0071】ホストコンピュータ1は、ステップf31の待機のあと、ステップf32でデジタル複合機2に対してバージョン情報の要求を行う。デジタル複合機2のバージョン情報送受信部25は、バージョン管理手段として、ステップf21の待機中にホストコンピュータ1からバージョン情報の要求を受け、以下ステップf22からステップf24までで、デバイスドライバ管理サーバーコンピュータ3から取得するバージョン情報を、ホストコンピュータ1に送信する。ホストコンピュータ1は、ステップf33で、デジタル複合機2からイントラネット6を介して送信されるデバイスドライバのバージョン情報を受信し、ステップf31に戻る。

【0072】本実施形態は、図2に示すように、イントラネット6に接続されている不特定多数のホストコンピュータ1にバージョン情報を送信する構成と比較して、たとえばアプリケーションプログラムの新たな導入の際にバージョン情報の送信を要求するホストコンピュータ1に対してのみバージョン情報を送信する。たとえば、現状のバージョンで充分であって、バージョンアップの

必要がないホストコンピュータ1は、バージョン情報を要求しないので、デバイスドライバ管理サーバーコンピュータ3から取得するバージョン情報の送信対象から除外し、ユーザネットワーク環境であるイントラネット6のトラフィックを減らすことができる。

【0073】図8は、本発明の実施の第7形態としてのコンピュータネットワークシステムの構成を示す。本実施形態で、ホストコンピュータ1、ゲートウェイ4、インターネット5およびイントラネット6はそれぞれ図1の実施形態と同等である。本実施形態で図1の実施形態と相違している点は、デジタル複合機42では内部にデバイスドライバの送受信を行うドライバ送受信部47と、デバイスドライバを保存する保存手段としてのドライバファイルライブラリ48が設けられていることである。イントラネット6やインターネット5を介してデバイスドライバの送受信を行うために、OS51やWWWサーバー/クライアント52も、デバイスドライバの送受信が可能ないようにしてある。デジタル複合機42が保持するデバイスドライバは、デバイスドライバ管理サーバーコンピュータ53から取得する。デバイスドライバ管理サーバーコンピュータ53には、デバイスドライバのバージョン情報だけでなく、デバイスドライバ自体をデジタル複合機42などの周辺機器に送信するデバイスドライバ送信部56が設けられている。OS61およびWWWサーバー62も、デジタル複合機42へのデバイスドライバの送信が可能ないように構成されている。

【0074】図9は、図8のシステム構成でのバージョン管理動作の手順を示す。本実施形態でのデジタル複合機42の動作は、図6に示す実施形態でのデジタル複合機2の動作に類似している。すなわち、デバイスドライバ管理サーバーコンピュータ53は、ステップg11で待機し、デジタル複合機42からのバージョン情報の要求があれば、ステップg12でデバイスドライバのバージョン情報の送信を行う動作を繰返す。デジタル複合機42は、ステップg21からステップg22、ステップg23およびステップg24の各ステップの動作を、図6の実施形態のステップe11、ステップe12、ステップe13およびステップe14の各ステップとそれぞれ同等に行う。本実施形態のステップg25では、デバイスドライバ管理サーバーコンピュータ53から受信したデバイスドライバのバージョン情報を更新し、ステップg26でデバイスドライバ管理サーバーコンピュータ53に対して、更新されたバージョンのデバイスドライバのダウンロードを要求する。ステップg27では、デバイスドライバ管理サーバーコンピュータ53から送信されるデバイスドライバを受信し、ステップg21に戻る。すなわち、ステップg26およびステップg27では、ダウンロード手段としてデバイスドライバをダウンロードする。デバイスドライバ管理サーバーコンピュータ53では、ステップg31の待機のあと、デジタル複

合機42からデバイスドライバのダウンロード要求に応じて、ステップg32でプリンタドライバファイル34またはスキャナドライバファイル35として格納されているデバイスドライバを、デジタル複合機2に送信して、ステップg31に戻る。

【0075】本実施形態では、デジタル複合機42には、常に最新のデバイスドライバが格納されていることになる。各ホストコンピュータ1は、デジタル複合機2からイントラネット6を介して最新のデバイスドライバをいつでもダウンロードすることができ、デバイスドライバのバージョンアップを容易に行うことができる。デバイスドライバのバージョンアップのために、各ホストコンピュータ1が直接デバイスドライバ管理サーバコンピュータ53にインターネット5を介してアクセスを行う必要がないので、デバイスドライバ管理サーバコンピュータ53へのトラフィックを減少させることができる。

【0076】図10は、本発明の実施の第8形態としての図8のネットワークシステムのバージョン管理手順を示す。本実施形態は、図9に示す実施形態に類似し、ステップh11でのバージョン情報の比較は、図9のステップg21からステップg23までの各ステップの動作後に、ステップg24でバージョン情報の比較を行う手順とそれぞれ同様に行う。ステップh11でバージョン情報が現在保持しているバージョンと同じでないと判断されたあとのステップh12、ステップh13およびステップh14の各ステップの動作は、図9のステップg25、ステップg26およびステップg27の各ステップとそれぞれ同等に行う。本実施形態では、次にステップh15で通知手段として、デジタル複合機42が各ホストコンピュータ1に対して、バージョンアップされたデバイスドライバをインストールするか否かのインストール要求についての問合せを行う。この問合せを受信するホストコンピュータ1のユーザは、最新のドライバが存在することを認識することができる。

【0077】図11は、本発明の実施の第9形態としての、図8のシステム構成でのバージョン管理の動作手順を示す。本実施形態では、デジタル複合機42の問合せ手段としてのバージョン情報送受信部25からデバイスドライバ管理サーバコンピュータ53に対するバージョン情報の問合せに、予め設定されているデバイスドライバ管理サーバコンピュータ53のアドレスへのログインパスワードを使用する。すなわち、デバイスドライバ管理サーバコンピュータ53は、動作開始後、ステップi11で、パスワードを待機して受信する。パスワードが受信されると、ステップi12でパスワードを判定し、判定結果を送信する。ステップi13で、ステップi12の判定結果でパスワードが正しくないときには、ステップi11に戻る。ステップi13でパスワードが正しいときには、ステップi14で、デバイスドライバ管理サーバコンピュータ53に格納されているデバイスドライバのバージョン情報を送信し、ステップi11に戻る。デジタル複合機42では、動作開始後、ステップi21で待機し、ステップi22では、デバイスドライバ管理サーバコンピュータ53のURLを指定してパスワードを送信する。このパスワードは、デバイスドライバ管理サーバコンピュータ53のURLに対応して、予め設定されているログインパスワードを用いる。ステップi23では、デバイスドライバ管理サーバコンピュータ53からのパスワード判定結果を受信する。ステップi24で、判定結果がOKでないときには、ステップi21の待機に戻る。ステップi24で、判定結果がOKであるときには、ステップi25で、デバイスドライバ管理サーバコンピュータ53から送信されるデバイスドライバのバージョン情報を受信する。

【0078】本実施形態では、デバイスドライバ管理サーバコンピュータ53側で、予め設定されているログインパスワードを有する周辺機器だけがアクセス可能となり、そのような周辺機器に接続されていない不特定多数のホストコンピュータ1はデバイスドライバ管理サーバコンピュータ53にアクセスすることができなくなる。これによって、デバイスドライバ管理に対するセキュリティを向上させることができる。

【0079】図12は本発明の実施の第10形態として、図8のシステム構成でのバージョン管理の動作を示す。本実施形態は、図11に示す実施形態に類似し、相違点は、予め設定されているログインパスワードを使ってアクセスを行ったあとで、バージョン情報とともに次回にアクセスする場合のログインパスワードを改めて設定することである。すなわち、図12のステップj11、ステップj12、ステップj13の各ステップは、図11のステップi11、ステップi12、ステップi13の各ステップとそれぞれ同等の動作を行う。デバイスドライバ管理サーバコンピュータ53は、ステップj13で、パスワードが等しいときに、ステップj14として、デバイスドライバ管理サーバコンピュータ53に格納されているデバイスドライバのバージョン情報および次回のパスワードを送信し、ステップj11に戻る。デジタル複合機52側の動作は、ステップj21、ステップj22、ステップj23およびステップj24の各ステップの動作が、図11のステップi21、ステップi22、ステップi23およびステップi24の各ステップとそれぞれ同等である。デジタル複合機42は、ステップj24で判定結果がOKであるとき、ステップj25でパスワード獲得手段として、デバイスドライバ管理サーバコンピュータ53から送信されるデバイスドライバのバージョン情報および次回パスワードを受信する。次にステップj26でパスワード保持手段として、次回パスワードを格納し、次回の問合せに備えて保持しておく。



【0080】本実施形態では、デバイスドライバ管理サーバーコンピュータ53は、各デジタル複合機42毎に個別に、かつ逐次変化していくパスワードを指定することもでき、そのようなパスワードを指定すれば、そのパスワードでのみデジタル複合機42はアクセス可能となる。デバイスドライバ管理サーバーコンピュータ53は、同一のパスワードでのアクセスを検知して、不正なアクセスと判断することもでき、何らかの原因でパスワードが第三者に知られるようなときでも、不正なアクセスを防止することができる。

【0081】図13は、本発明の実施の第1形態としての、図8のネットワークシステムのバージョン管理動作を示す。ただし、本実施形態では、デバイスドライバが複数のファイルに分けて構成され、デバイスドライバを構成するファイルは、複数台設置されているデバイスドライバ管理サーバーコンピュータ53に分散して準備されているものとする。これに対応してデジタル複合機42のバージョン情報送受信部25は、アドレス保持手段として、複数のデバイスドライバ管理サーバーコンピュータ53のアドレスを保持している。ダウンロード手段として、デバイスドライバをダウンロードする際には、最初のデバイスドライバ管理サーバーコンピュータ53から順次アクセスして、デバイスドライバを構成するファイルを順次ダウンロードする。したがって、デバイスドライバ管理サーバーコンピュータ53を分散させることができ、バージョンアップ時などに特定のデバイスドライバ管理サーバーコンピュータ53にダウンロードのためのアクセスが集中してトラフィックが重くなるのを軽減し、デバイスドライバのバージョンアップに伴うネットワークを介してのデバイスドライバの配布を高速に行うことができる。

【0082】図13では、デジタル複合機42が、バージョンアップされたデバイスドライバをダウンロードする手順を示す。ステップk11で、デジタル複合機42のバージョン情報送受信部25は、ダウンロードの対象となるデバイスドライバを複数のファイルに分けて管理しているデバイスドライバ管理サーバーコンピュータ53の数を獲得する。次にステップk12で、デバイスドライバを構成するファイルのうちの最初のファイルを管理しているデバイスドライバ管理サーバーコンピュータ53のアドレスを獲得する。デバイスドライバ管理サーバーコンピュータ53のアドレスは、デジタル複合機42の製造メーカなどから、予め指定される。次にステップk13で、指定されているアドレスのデバイスドライバ管理サーバーコンピュータ53から、そのデバイスドライバ管理サーバーコンピュータ53で保持しているデバイスドライバファイルをダウンロードする。ステップk14では、次のファイルを管理しているデバイスドライバ管理サーバーコンピュータ53のアドレスを獲得する。ステップk15では、ファイルをダウンロードした

デバイスドライバ管理サーバーコンピュータ53が、ステップk11で獲得したデバイスドライバ管理サーバーコンピュータ53の数まで到達して、最終のデバイスドライバ管理サーバーコンピュータ53であるか否かを判断する。最終でないと判断されるときには、ステップk13に戻って、ファイルのダウンロードを継続する。

【0083】ステップk14で獲得する次のデバイスドライバ管理サーバーコンピュータ53のアドレスは、ステップk11で獲得するデバイスドライバ管理サーバーコンピュータ53の数や、ステップk12で獲得する最初のデバイスドライバ管理サーバーコンピュータ53のアドレスと同様に、予め周辺機器の製造メーカなどで指定しておくことができる。また、デバイスドライバを複数のファイルに分割するか否かは、実際にデバイスドライバ管理サーバーコンピュータ53にアクセスして判明するようにしてもよい。すなわち、周辺機器の製造メーカなどは、1つのデバイスドライバ管理サーバーコンピュータ53のみを指定しておき、ユーザ側からそのデバイスドライバ管理サーバーコンピュータ53にアクセスしてデバイスドライバをダウンロードした結果で判明するようにしてもよい。

【0084】デバイスドライバが複数のファイルに分割されて複数のデバイスドライバ管理サーバーコンピュータ53から分散してダウンロードする必要がある場合には、アクセスしたデバイスドライバ管理サーバーコンピュータ53が最初のデバイスドライバ管理サーバーコンピュータ53となり、デバイスドライバとして構成するのに必要なファイルを保持するデバイスドライバ管理サーバーコンピュータ53の数やそのアドレスを、最初のデバイスドライバ管理サーバーコンピュータ53から周辺機器に知らせるようにすることもできる。さらに、複数のデバイスドライバ管理サーバーコンピュータ53は、デバイスドライバを構成するファイルの順番に、先行するデバイスドライバ管理サーバーコンピュータ53からファイルをダウンロードする際に、後続のデバイスドライバ管理サーバーコンピュータ53のアドレスを指定するようにすることもできる。最終のデバイスドライバ管理サーバーコンピュータ53は、後続のデバイスドライバ管理サーバーコンピュータ53のアドレスを指定しないことによって、最終のデバイスドライバ管理サーバーコンピュータ53であることを知らせることができる。

【0085】図14は、本発明の実施の第12形態として、図8に示すネットワークシステムでデバイスドライバのバージョン管理を行う手順を示す。本実施形態では、デバイスドライバ管理サーバーコンピュータ53に、ホストコンピュータ1のOS環境を調査するプログラムが格納されており、デジタル複合機42はホストコンピュータ1からの要求に応じて、デバイスドライバ管理サーバーコンピュータ53にそのOS環境調査プログ

ラムを要求して獲得することができる。デジタル複合機42がOS環境調査プログラムを獲得すると、ホストコンピュータ1へ送信し、OS環境の調査を行わせ、調査結果を受信してデバイスドライバ管理サーバーコンピュータ53に送信する。デバイスドライバ管理サーバーコンピュータ53からは、OS環境調査プログラムの調査結果に対応したデバイスドライバをダウンロードすることができる。

【0086】すなわち、図14に示すように、デバイスドライバ管理サーバーコンピュータ53は、ステップ111で待機し、デジタル複合機42からの要求があれば、ステップ112でOS環境調査プログラムを送信して、再びステップ111の待機に戻る。デバイスドライバの送信に関連しては、ステップ121で待機して、OS環境調査プログラムの調査結果が送信されれば受信し、その結果に応じて、ステップ122でデバイスドライバを送信し、ステップ121の待機に戻る。

【0087】デジタル複合機42では、ステップ131での待機に続いて、ホストコンピュータ1からのOS調査環境プログラムの要求に応じて、調査プログラム獲得手段として動作する。すなわち、ステップ132でデバイスドライバ管理サーバーコンピュータ53にOS環境調査プログラムを要求し、ステップ133では調査プログラム送付手段として、デバイスドライバ管理サーバーコンピュータ53からOS環境調査プログラムを受信し、ホストコンピュータ1に送信する。ステップ134では、調査結果獲得手段としてホストコンピュータ1からの調査結果を受信し、調査結果送付手段としてデバイスドライバ管理サーバーコンピュータ53に送信し、ダウンロード手段としてデバイスドライバ管理サーバーコンピュータ53から調査結果に対応したデバイスドライバをダウンロードして、ステップ131に戻る。ホストコンピュータ1は、ステップ141での待機のあと、ステップ142で、OS環境調査プログラムの要求をデジタル複合機42に対して行う。ステップ143では、デジタル複合機42からOS環境調査プログラムを受信し、ステップ144で、OS環境調査プログラムを実行した調査結果をデジタル複合機42に送信して、ステップ141に戻る。

【0088】本実施形態では、ホストコンピュータ1に対してOS環境調査プログラムの調査結果に対応するデバイスドライバが、デジタル複合機42のデバイスドライバファイルライブラリ48に格納され、インターネット6を介していつでもホストコンピュータ1にダウンロードしてインストールすることができるので、ホストコンピュータ1のハードウェアやソフトウェアに適合したデバイスドライバによるバージョンアップが可能となる。また、OS環境調査プログラムを、周辺機器であるデジタル複合機42ではなく、デバイスドライバ管理サーバーコンピュータ53で管理し、デバイスドライバ管

理サーバーコンピュータ53から取得することができる。これによって、ホストコンピュータ1のOS11を全面的に他のOSに変更するような場合であっても、デバイスドライバ管理サーバーコンピュータ53にそのOSに適合したOS環境調査プログラムをデバイスドライバ管理サーバーコンピュータ53で準備しておけば、新たなOSを搭載したホストコンピュータ1でも十分な環境調査が可能なOS環境調査プログラムを用意することができる。

【0089】図15は、本発明の実施の第13形態としての図8のネットワークシステムによるバージョン管理の手順を示す。本実施形態で、デバイスドライバ管理サーバーコンピュータ53には、複数バージョンのデバイスドライバが格納されており、デジタル複合機42のバージョン情報送受信部25はバージョン情報獲得手段として、すべての種類のバージョン情報を獲得して、ホストコンピュータ1に送信し、ホストコンピュータ1から要求されたデバイスドライバのみをダウンロードする。すなわちステップm11では、デバイスドライバ管理サーバーコンピュータ53に問合せを行って、格納されているデバイスドライバの全バージョン情報を獲得する。ステップm12では、獲得したデバイスドライバの全バージョンの情報をホストコンピュータ1に送信する。ステップm13では、バージョン選択手段として、ホストコンピュータ1からインストールが要求されたバージョン情報に対応するデバイスドライバを選択し、ダウンロード手段としてデバイスドライバ管理サーバーコンピュータ53からダウンロードする。

【0090】デバイスドライバは、バージョンアップによって機能が追加されることが一般的である。しかしながら、ホストコンピュータ1でのハードウェアやソフトウェアの環境上、必ずしも追加された機能が必要でない場合もあり得る。機能が追加されたデバイスドライバは、プログラムサイズも大きくなるので、ホストコンピュータ1のメモリもバージョンアップによる追加で多く必要となってしまう。このため、バージョンアップも、複数の要求に応じて、一部のバージョンでは、機能の追加を抑えてプログラムサイズを重視するように行うことができる。複数のバージョンが存在するときには、ホストコンピュータ1でインストールさせるバージョンを選択することが可能なようにして、広範囲のホストコンピュータ1への対応を可能にすることができる。

【0091】図16は、本発明の実施の第14形態として、図8のネットワークシステムでのバージョン管理の動作手順を示す。本実施形態は、図15の実施形態に類似し、相違点は、デジタル複合機42で、デバイスドライバ管理サーバーコンピュータ53に格納されているすべてのデバイスドライバがそれぞれ対応しているOS情報を要求し、そのすべてのデバイスドライバを予めダウンロードして保存しておくことである。

【0092】すなわち、図16のステップn11では、デジタル複合機42は、デバイスドライバ管理サーバーコンピュータ53に格納されているデバイスドライバが対応しているすべてのOS情報を獲得する。ステップn12では、各対応OS情報毎に、デバイスドライバをダウンロードする。ステップn13で、すべてのOS情報に対応したデバイスドライバのダウンロードが終了しているか否かを判断する。終了していないときには、ステップn12に戻る。本実施形態では、デジタル複合機42にイトラネット6を介して接続されているすべてのホストコンピュータ1で使用されているOSに対応するデバイスドライバを、予めデジタル複合機42にダウンロードして保存しておくので、必要が生じたデバイスドライバだけを逐次ダウンロードする場合に比べて、デバイスドライバ管理サーバーコンピュータ53へのアクセスによるトラフィックを軽減することができる。

【0093】図17は、本発明の実施の第15形態として、図8のネットワークシステムでのバージョン管理の動作手順を示す。本実施形態は、図9に示す実施形態で、デバイスドライバをデバイスドライバ管理サーバーコンピュータ3からデジタル複合機2にダウンロードする動作の部分に類似する。すなわち、デバイスドライバ管理サーバーコンピュータ3は、暗号処理されたデバイスドライバを、プリンタドライバファイル34やスキャナドライバファイル35に格納しておく。ステップo11での待機のアと、デジタル複合機2、42からの要求があれば、ステップo12で、格納されている暗号処理されたデバイスドライバファイルを送信し、ステップo11に戻る。デジタル複合機2、42では、ステップo21で、暗号処理されたデバイスドライバのダウンロードの要求をデバイスドライバ管理サーバーコンピュータ3、53に対して行う。ステップo22では、デバイスドライバ管理サーバーコンピュータ3、53から暗号処理されたデバイスドライバを受信する。ステップo23で、デジタル複合機53では、暗号解読手段として暗号を解読し、その状態でデバイスドライバをドライバファイルライブラリ48に格納する。

【0094】暗号化処理では、データの内容は不明であっても、データとしてエラーが生じているか否かなどは解るようになっているので、暗号処理されたデバイスドライバの内容が、第三者によって不所望に変更されているか否かは容易に判断することができる。デジタル複合機42のドライバ送受信部47が暗号解読の機能を備え、第三者によって不所望に変更されていないことが保証されたデバイスドライバを解読してドライバファイルライブラリ48に格納することによって、ホストコンピュータ1では解読されたデバイスドライバをドライバファイルライブラリ48からダウンロードして利用することができ、暗号を解読するプログラムなどを必要としない。

【0095】図18は、本発明の実施の第16形態として、図8に示すネットワークシステムでのバージョンアップの動作手順を示す。本実施形態で、デバイスドライバ管理サーバーコンピュータ53には、プリンタドライバ12やスキャナドライバ13などの周辺機器の構成毎に区分して、プリンタドライバファイル34やスキャナドライバファイル35などのように分けて格納されている。周辺機器は、その構成を示す情報をデバイスドライバ管理サーバーコンピュータ53に送信して、デバイスドライバ管理サーバーコンピュータ53は、その構成情報に対応したファイルリストを送信し、周辺機器はそのファイルリストに沿って、順次必要なデバイスドライバをダウンロードする。

【0096】すなわち、問合せ手段としてのバージョン情報送受信部25は、構成情報送信手段として、ステップp11では、たとえばデジタル複合機42にはプリンタとスキャナとが含まれていることを示す構成情報を、デバイスドライバ管理サーバーコンピュータ53に送信する。ステップp12では、ファイルリスト受信手段として、デバイスドライバ管理サーバーコンピュータ53から、構成に対応しているファイルリストを受信する。ステップp13では、構成に対応した全デバイスドライバファイルを、ファイルリストに沿ってダウンロードする。

【0097】本実施形態で、デバイスドライバ管理サーバーコンピュータ53は、周辺機器毎にデバイスドライバを格納しておくのではなく、周辺機器の構成毎に格納しておく。たとえば複写機とプリンタとで、プリンタ部分のデバイスドライバは共通であり、複写機にはさらにスキャナ部分のデバイスドライバが必要となる。デバイスドライバ管理サーバーコンピュータ53に、複写機に対してプリンタ部分とスキャナ部分とのデバイスドライバを備え、プリンタに対してはプリンタ部分のみのデバイスドライバを備えるといったように、周辺機器毎にデバイスドライバを格納しておく、プリンタ部分のように重複して格納する必要が生じる。プリンタ部分とスキャナ部分とのように、周辺機器の構成毎にデバイスドライバを格納しておけば、デバイスドライバ管理サーバーコンピュータ53には重複したデバイスドライバは格納されなくなり、格納のために必要なデータ量を大幅に削減することができる。デバイスドライバ管理サーバーコンピュータ53では、デバイスドライバをハードディスクなどに格納しておくので、周辺機器毎ではなく、周辺機器の構成毎にデバイスドライバを格納することによって、ハードディスクなどの節約を図ることができる。

【0098】図19は、本発明の実施の第17形態としての図8のネットワークシステムでのデバイスドライバの管理手順を示す。本実施形態は、図15に示す実施形態に類似し、相違点は、デジタル複合機42のバージョン情報送受信部25は、問合せ手段として、デバイスド

ライバ管理サーバーコンピュータ53に格納されているすべてのデバイスドライバがそれぞれ対応しているOS情報を要求して獲得し、それをインストール可能OS情報としてホストコンピュータ1に送信することである。ホストコンピュータ1からインストール要求があったデバイスドライバのみをデバイスドライバ管理サーバーコンピュータ53からデジタル複合機42にダウンロードする点である。すなわち、ステップq11では、デバイスドライバ管理サーバーコンピュータ53に格納されているデバイスドライバに対応しているOS情報を獲得する。ステップq12で、デジタル複合機42からホストコンピュータ1に対して、デバイスドライバがデバイスドライバ管理サーバーコンピュータ53で管理されていて、インストール可能なOS情報を通知する。ホストコンピュータ1は、デジタル複合機42から通知されたOS情報を参照し、デバイスドライバのインストールを要求するか否かを判断する。ホストコンピュータ1から、デバイスドライバのインストールが要求されると、ステップq13で、デジタル複合機42は、インストール要求されたOS情報に対応するデバイスドライバを、デバイスドライバ管理サーバーコンピュータ53からダウンロードする。

【0099】本実施形態では、デバイスドライバ管理サーバーコンピュータ53側で、ホストコンピュータ1にインストールするデバイスドライバに対応するOS情報とバージョン情報とを把握することができ、インターネット5を介して接続されるホストコンピュータ1で、使用頻度の高いOSなどについての情報も得ることができるので、使用頻度が高く、重点的に開発を進める必要があるOSを割出すこともできる。

【0100】図20は、本発明の実施の第18形態として、図8のネットワークシステムによるデバイスドライバの管理動作を示す。本実施形態は、図9に示す実施形態に類似し、相違点は、デバイスドライバのダウンロードを、デジタル複合機42に対して、ホストコンピュータ1からのアクセスがない時点で行うことである。すなわち、ステップr11までの動作は、たとえば図9のステップg21、ステップg22、ステップg23およびステップg24と同等に行い、ステップr12でホストコンピュータ1からのアクセスが所定時間なくなるまで待機する。所定時間経過してもアクセスが無ければ、アクセス無し検知手段として動作し、待機を終了させる。ステップr13、ステップr14およびステップr15までの各ステップで、図9のステップg25、ステップg26およびステップg27の各ステップとそれぞれ同等の動作を行う。デバイスドライバ管理サーバーコンピュータ53側でのステップr21およびステップr22の各ステップの動作も、図9のステップg31およびステップg32の各ステップの動作と同様である。

【0101】本実施形態では、ステップr11で、デバ

イスドライバ管理サーバーコンピュータ53から受信したデバイスドライバのバージョン情報が現在保持しているバージョンと異なると判断されても、直ちにダウンロードの要求を発生するのではなく、各ホストコンピュータ1からのアクセスが所定時間にわたってなくなるまで待機し、アクセスがなくなってからバージョン情報の更新からデバイスドライバのダウンロードまでを行う。したがって、イントラネット6などのユーザネットワーク環境のトラフィックが少ない状態でデバイスドライバのダウンロードを開始することができ、高速でデバイスドライバをダウンロードすることができる。

【0102】以上説明した各実施形態は、基本的な考え方の異なる部分を除き、組合せて用いることもできる。また、周辺機器として、デジタル複合機2、42を用いる場合について説明しているけれども、単独のプリンタやスキャナなどを周辺機器として用いることもできる。

【0103】

【発明の効果】以上のように本発明によれば、ネットワークを介して複数の情報処理装置によって利用される周辺機器は、各情報処理装置で周辺機器を利用するためのデバイスドライバを、外部のネットワークを介して接続される管理サーバー装置へバージョン情報を問合せ、管理サーバー装置から得られるバージョン情報を内部のネットワークを介して接続される情報処理装置に通知するので、情報処理装置のユーザは現在インストールされているデバイスドライバが最新のバージョンであるか否かを容易に判定することができる。

【0104】また本発明によれば、周辺機器が外部のネットワークを介して管理サーバー装置に対して、デバイスドライバのバージョン情報を一定時間間隔で問合せ、各情報処理装置のユーザは、インストールされているデバイスドライバが最新のバージョンであるか否かを常に監視することができる。

【0105】また本発明によれば、管理サーバー装置側から得られるバージョン情報を問合せすべき間隔に基づいて、周辺機器の問合せ手段が管理サーバー装置に対してデバイスドライバのバージョン情報の問合せを行うので、管理サーバー装置側でのバージョンアップのスケジュールに合わせた問合せを行うことができ、無駄な問合せを回避して、管理サーバー装置へのトラフィックを軽減することができる。

【0106】また本発明によれば、電源投入後のウォーミングアップ中で、使用される可能性が少ないタイミングに、周辺機器の問合せ手段は、外部のネットワークを介して管理サーバー装置に対するバージョン情報の問合せを行うことができるので、内部のネットワークを介して接続される情報処理装置からの周辺機器の使用の要求に対して、バージョン情報の問合せ中であることに基づく遅延を避けることができる。電源の立ち上り検知は比較的簡単に行うことができるので、一定時間間隔で問合せ

せるような構成に比較して、コストダウンも可能となる。

【0107】また本発明によれば、周辺機器に内部のネットワークを介して接続される情報処理装置が使用しているデバイスドライバのバージョン情報を周辺機器で管理しておき、管理サーバー装置に保存されているデバイスドライバのバージョン情報との相違がある場合にのみ、情報処理装置に対する通知が行われるので、内部のネットワーク環境でのトラフィックを軽減することができる。

【0108】また本発明によれば、周辺機器は、内部のネットワークを介して接続される情報処理装置のうち、バージョン情報の通知を要求している情報処理装置にのみバージョン情報を通知させるので、内部のネットワーク環境でのトラフィックを減らすことができる。

【0109】また本発明によれば、周辺機器は、管理サーバー装置からデバイスドライバをダウンロードして保存するので、内部のネットワークを介して接続される情報処理装置は、管理サーバー装置に対してアクセスしてデバイスドライバをダウンロードする必要はなく、内部のネットワークを介して周辺機器から必要なデバイスドライバを取得することができる。周辺機器の保存手段には、バージョン情報の比較に基づいて、バージョン情報に差異が生じたときのみ管理サーバー装置からデバイスドライバをダウンロードするので、常に最新のデバイスドライバを保存しておくことができる。

【0110】また本発明によれば、周辺機器の通知手段は、内部のネットワークを介して接続される情報処理装置に対して、ダウンロードされたデバイスドライバのインストールを実行するか否かを問合せるので、情報処理装置のユーザは、周辺機器に最新のドライバがダウンロードされて存在していることを認識し、最新のバージョンのデバイスドライバをインストールするか否かを判断することができる。

【0111】また本発明によれば、デバイスドライバを複数のファイルに分けて、複数の管理サーバー装置で管理するので、周辺機器にデバイスドライバをダウンロードする際に、分散された複数の管理サーバー装置からデバイスドライバを分けてダウンロードすることができる。デバイスドライバのバージョンアップ時には、管理サーバー装置にアクセスが集中しやすいので、デバイスドライバを複数の管理サーバー装置で分散して管理することによって、特定の管理サーバー装置へのアクセスの集中を避け、トラフィックの軽減を図ることができる。

【0112】また本発明によれば、周辺機器の問合せ手段が、外部のネットワークに接続される管理サーバー装置にアクセスする際には、管理サーバー装置側で予め設定されているログインパスワードを使用するので、ログインパスワードを有する周辺機器だけがアクセス可能となり、セキュリティを向上させることができる。

【0113】また本発明によれば、周辺機器の問合せ手段が管理サーバー装置にアクセスする際に用いるログインパスワードを、管理サーバー装置から指定することができるので、連続して同一のパスワードを使用してのアクセスなどを容易に検知することができ、不正なアクセスを有効に防止することができる。

【0114】また本発明によれば、管理サーバー装置から提供されるOS環境調査プログラムで、周辺機器は内部のネットワークを介して接続される情報処理装置についてのOS環境の調査結果を獲得して、獲得された調査結果を管理サーバー装置に送付するので、管理サーバー装置からはOS環境調査プログラムの調査結果に基づいたデバイスドライバを取得することが可能になり、各情報処理装置の使用環境に適合したデバイスドライバでのバージョンアップが可能となる。また、OS環境調査プログラムを、管理サーバー装置から提供するので、新たなOSを情報処理装置に搭載するような場合でも、容易に対応させることができる。

【0115】また本発明によれば、デバイスドライバがバージョンアップして、複数のバージョンのデバイスドライバが存在するときに、周辺機器は、内部のネットワークを介して接続される複数の情報処理装置に対し、各情報処理装置のシステムに適合したバージョンのデバイスドライバを選択してインストールさせることができ、広範囲の情報処理装置に対して対応することが可能となる。

【0116】また本発明によれば、周辺機器の保存手段には、複数のOS用のデバイスドライバを格納するので、周辺機器に対して内部のネットワークを介して接続される情報処理装置が、複数のOSを使用している場合でも、必要なデバイスドライバを周辺機器から提供することができる。広範なOSに対して、内部のネットワークのみでデバイスドライバの管理を行うことができ、外部のネットワークを介する管理サーバー装置への問合せや、管理サーバー装置からのダウンロードの回数を減らして、管理サーバー装置のトラフィックを軽減することができる。

【0117】また本発明によれば、管理サーバー装置から暗号化されたデバイスドライバをダウンロードして周辺機器内で暗号を解読するので、ダウンロードされたデバイスドライバの内容が変更されていないことが保証される。内部のネットワークを介して接続される情報処理装置では暗号の解読を行う必要がないようにして、最新のバージョンのデバイスドライバでも、確実かつ迅速に各情報処理装置に転送して、インストールすることができる。

【0118】また本発明によれば、周辺機器の構成情報に基づいて、管理サーバー装置からファイルリストとしてデバイスドライバが得られるので、管理サーバー装置側では周辺機器から送信される構成情報に基づいてファ

イルリストを作成すればよく、予め周辺機器の構成に応じたデバイスドライバを格納しておく必要がなくなり、管理サーバー装置での記憶容量の低減を図ることができる。

【0119】また本発明によれば、問合せ手段は、管理サーバー装置に問合せを行う際に、内部のネットワークを介して接続される情報処理装置にインストールされているデバイスドライバのOS情報とバージョン情報とを通知するので、管理サーバー装置側では、情報処理装置で使用頻度が高いOSを知ることができ、重点的にデバイスドライバの開発を進める必要があるOSを特定することができる。

【0120】また本発明によれば、内部のネットワークを介して接続される複数の情報処理装置からのアクセスが所定時間ないことを検知したときに、デバイスドライバのダウンロードを開始するので、ユーザによるネットワークを介しての周辺機器の利用が少ない状態で、外部のネットワークを介する管理サーバー装置からのデバイスドライバのダウンロードを行い、高速でデバイスドライバのダウンロードを行うことができる。

【0121】また本発明によれば、周辺機器の問合せ手段は、管理サーバー装置に対してデバイスドライバのバージョン情報の問合せをHTTPをプロトコルとして用いて行うので、内部のネットワークに関するネットワーク管理者のみならず、内部のネットワークと外部のネットワークとの間に設けられるゲートウェイに導入されているファイアウォールなどのセキュリティシステムを通過して、外部のネットワークから管理サーバー装置に対してバージョン情報の問合せを行うことができる。

【0122】また本発明によれば、周辺機器から内部のネットワークを介して接続される情報処理装置で行う要求や通知は、HTTPをプロトコルとして用いて行うので、外部のネットワークを介しての管理サーバー装置への問合せと、内部のネットワークを介しての情報処理装置への通知とを、同じプロトコルで処理することができ、周辺機器に対して管理サーバー装置と情報処理装置とをシームレスに接続することが可能になる。

【0123】さらに本発明によれば、周辺機器メーカーなどに設置され、常に最新のバージョンのデバイスドライバが保存されている管理サーバー装置に対して、周辺機器は内部のネットワークを介して接続される情報処理装置に代えて、デバイスドライバのバージョン情報を、外部のネットワークを介して問合せを行う。問合せの結果を、内部のネットワークを介して情報処理装置に通知することによって、情報処理装置のユーザは、情報処理装置が使用しているデバイスドライバが最新のバージョンのものであるか否かを判定することができる。

【図面の簡単な説明】

【図1】本発明の実施の第1形態としての概略的なシステム構成を示すブロック図である。

【図2】図1の実施形態でのデバイスドライバのバージョン管理手順を示すフローチャートである。

【図3】本発明の実施の第2形態としてのデバイスドライバのバージョン管理手順を示すフローチャートである。

【図4】本発明の実施の第3形態としてのデバイスドライバのバージョン管理手順を示すフローチャートである。

【図5】本発明の実施の第4形態としてのデバイスドライバのバージョン管理手順を示すフローチャートである。

【図6】本発明の実施の第5形態としてのデバイスドライバのバージョン管理手順を示すフローチャートである。

【図7】本発明の実施の第6形態としてのデバイスドライバのバージョン管理手順を示すフローチャートである。

【図8】本発明の実施の第7形態としてのシステム構成を示すブロック図である。

【図9】図8の実施形態でのデバイスドライバのバージョン管理手順を示すフローチャートである。

【図10】本発明の実施の第8形態としてのデバイスドライバのバージョン管理手順を示すフローチャートである。

【図11】本発明の実施の第9形態としてのデバイスドライバのバージョン管理手順を示すフローチャートである。

【図12】本発明の実施の第10形態としてのデバイスドライバのバージョン管理手順を示すフローチャートである。

【図13】本発明の実施の第11形態としてのデバイスドライバのバージョン管理手順を示すフローチャートである。

【図14】本発明の実施の第12形態としてのデバイスドライバのバージョン管理手順を示すフローチャートである。

【図15】本発明の実施の第13形態としてのデバイスドライバのバージョン管理手順を示すフローチャートである。

【図16】本発明の実施の第14形態としてのデバイスドライバのバージョン管理手順を示すフローチャートである。

【図17】本発明の実施の第15形態としてのデバイスドライバのバージョン管理手順を示すフローチャートである。

【図18】本発明の実施の第16形態としてのデバイスドライバのバージョン管理手順を示すフローチャートである。

【図19】本発明の実施の第17形態としてのデバイスドライバのバージョン管理手順を示すフローチャートである。

35

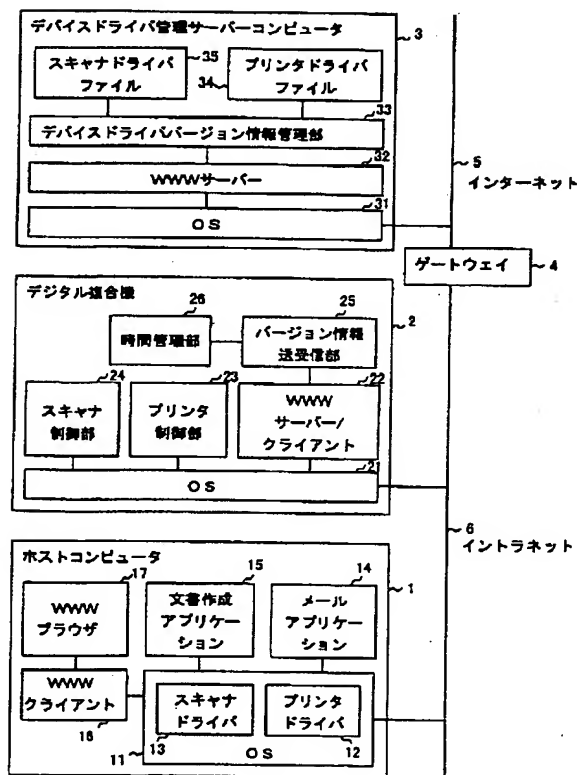
ある。

【図20】本発明の実施の第1形態としてのデバイスドライバのバージョン管理手順を示すフローチャートである。

【符号の説明】

- 1 ホストコンピュータ
- 2, 42 デジタル複合機
- 3, 53 デバイスドライバ管理サーバーコンピュータ
- 4 ゲートウェイ
- 5 インターネット
- 6 イントラネット
- 11, 21, 31, 51, 61 OS
- 12 プリンタドライバ
- 13 スキャナドライバ
- 14 メールアプリケーション

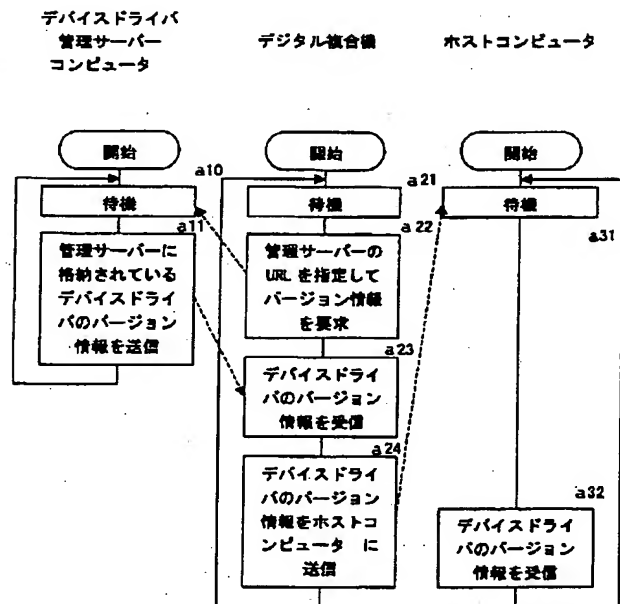
【図1】



36

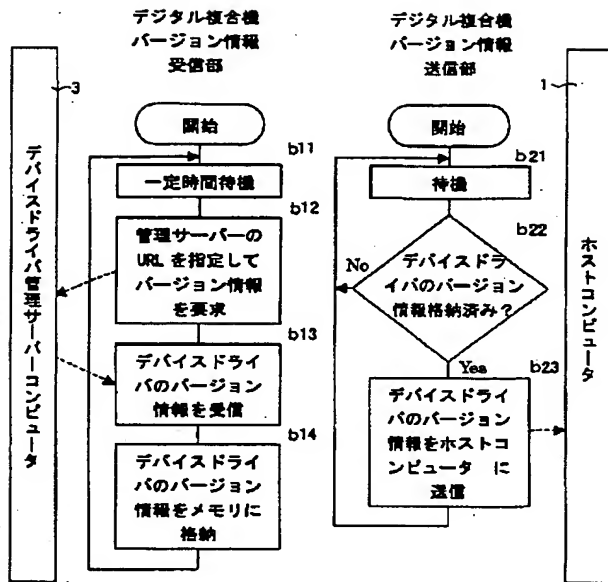
- 15 文書作成アプリケーション
- 16 WWWクライアント
- 17 WWWブラウザ
- 22, 52 WWWサーバー/クライアント
- 23 プリンタ制御部
- 24 スキャナ制御部
- 25 バージョン情報送受信部
- 26 時間管理部
- 32, 62 WWWサーバー
- 10 33 デバイスドライババージョン情報管理部
- 34 プリンタドライバファイル
- 35 スキャナドライバファイル
- 47 ドライバ送受信部
- 48 ドライバファイルライブラリ

【図2】

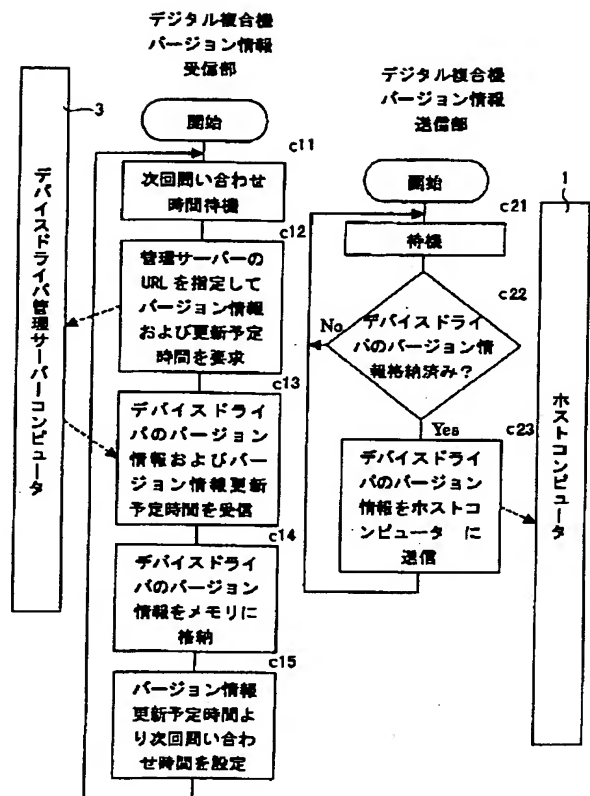




【図3】

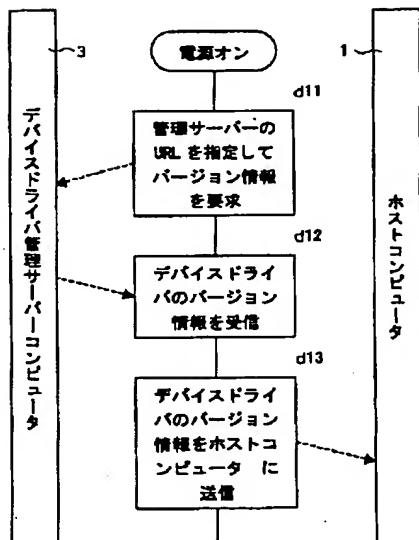


【図4】

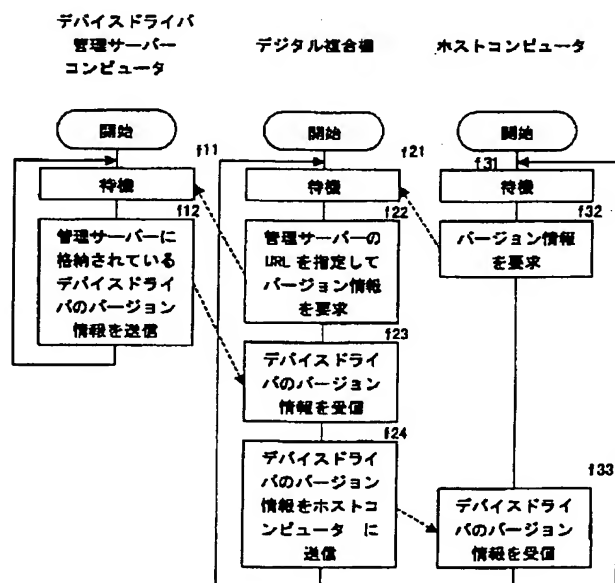


【図5】

デジタル複合機

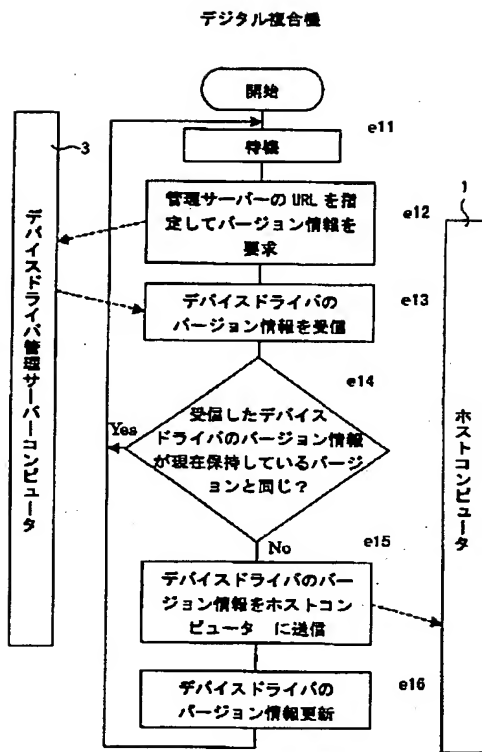


【図7】

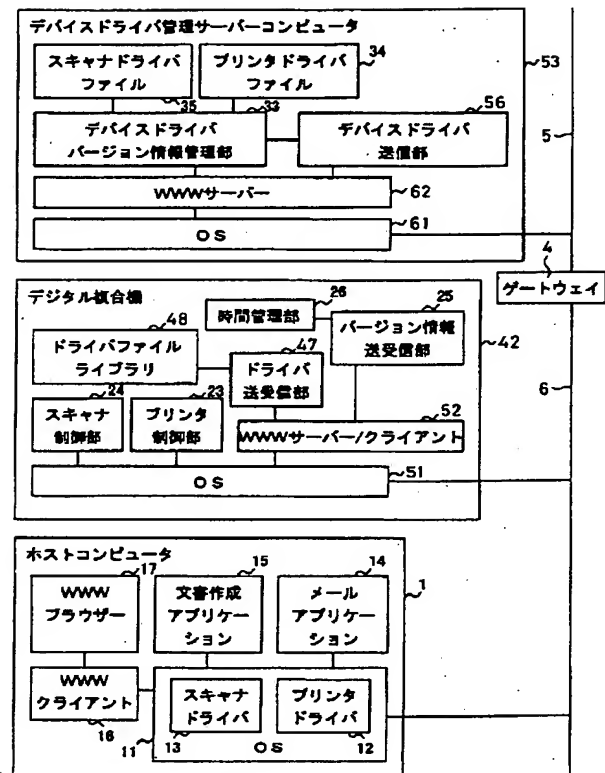




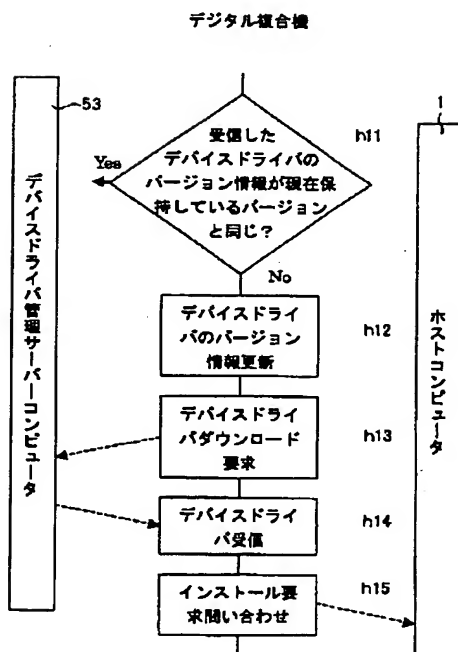
【図6】



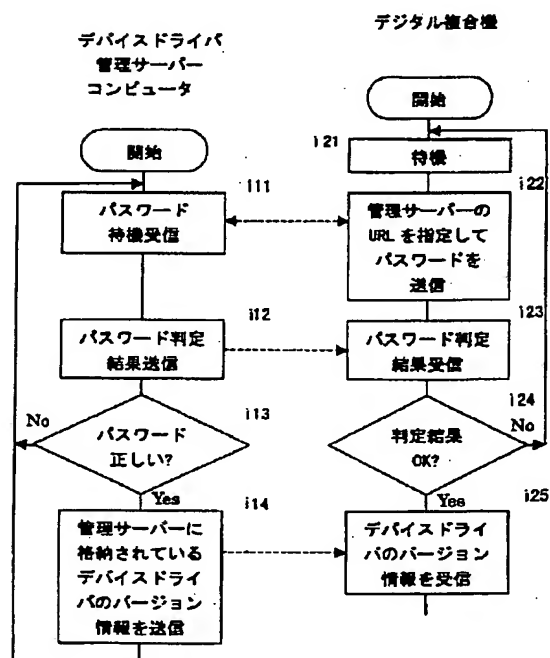
【図8】



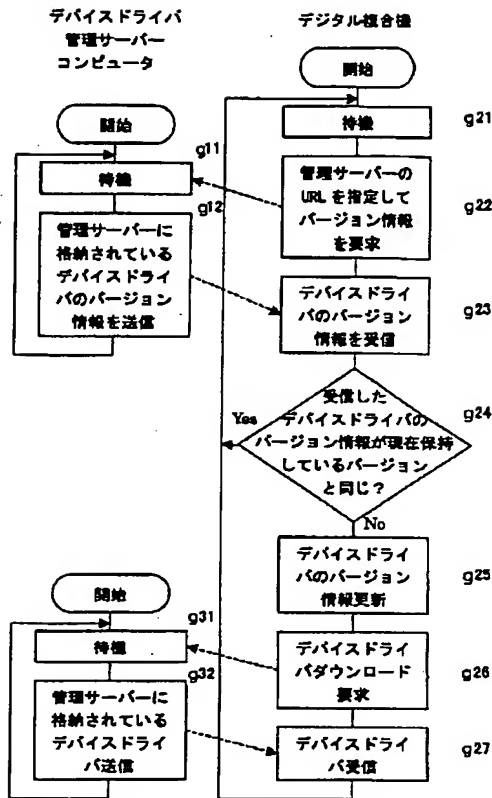
【図10】



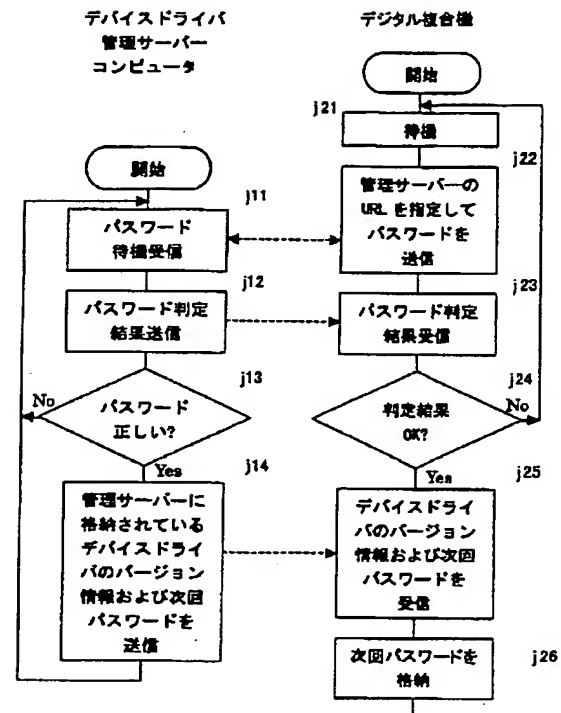
【図11】



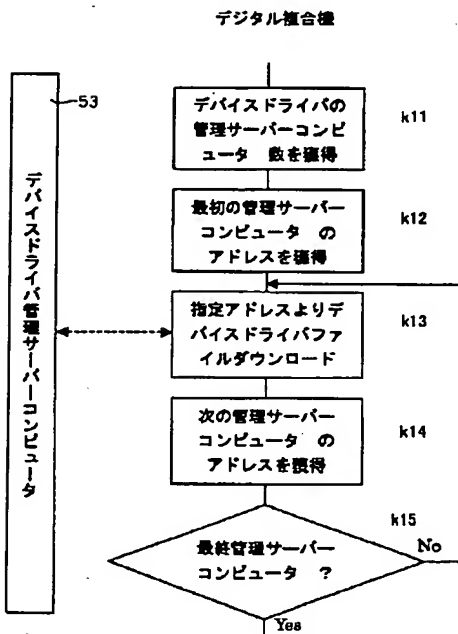
【図9】



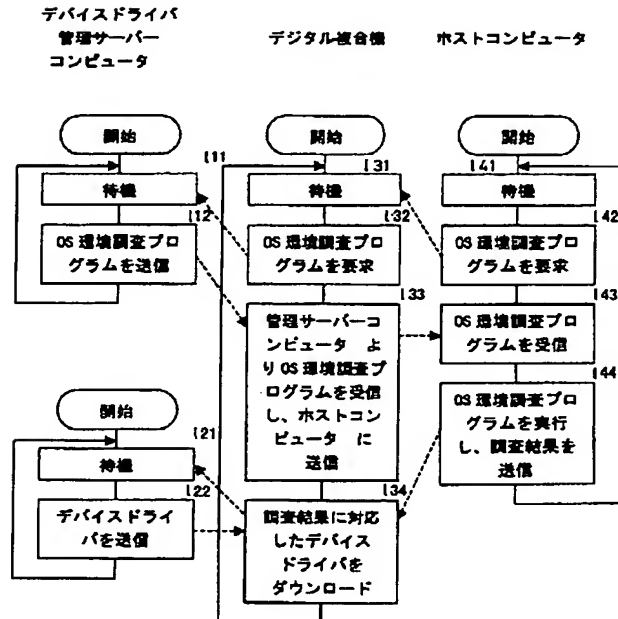
【図12】



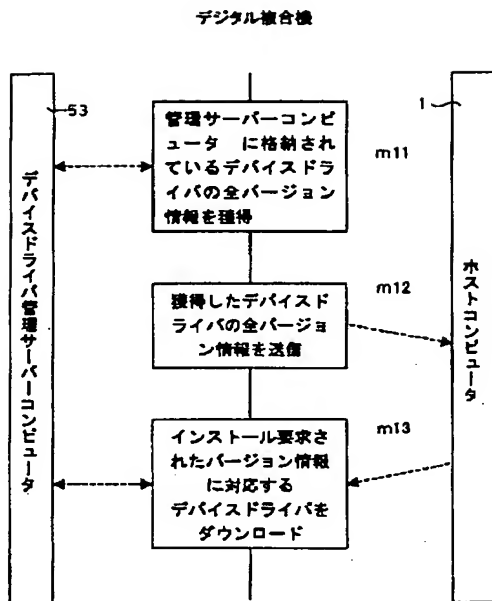
【図13】



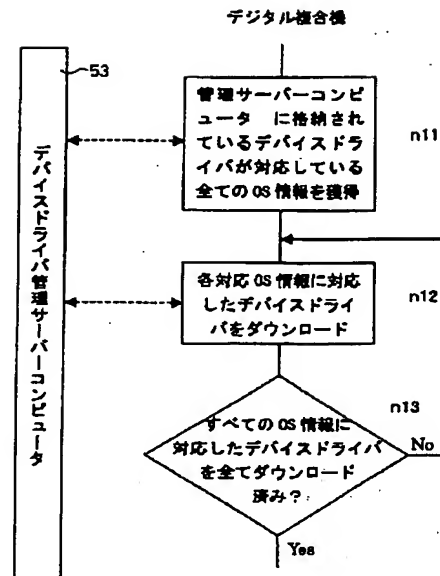
【図14】



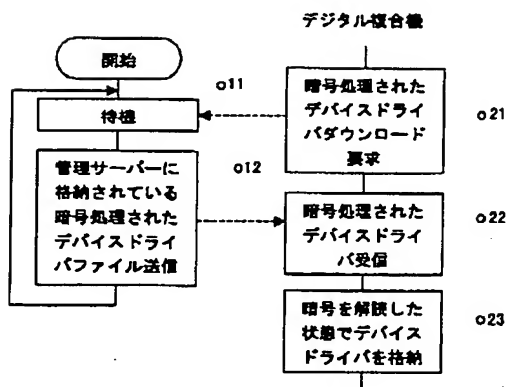
【図15】



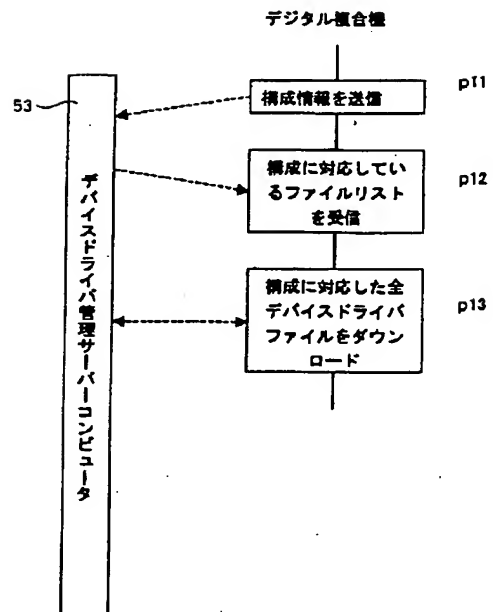
【図16】



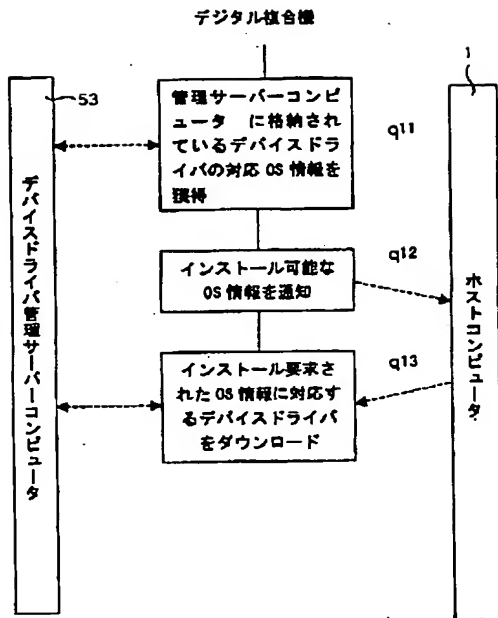
【図17】



【図18】



【図19】



【図20】

